Analytische benadering van rekenstrategieën

J. Klep SLO, Enschede

samenvatting

Binnen het ISWA-project van de SLO wordt een computerprogramma ontwikkeld ('Aritmeticus'), dat in staat is (informele) rekenstrategieën van kinderen te herkennen. Het programma maakt redeneringen waartoe een leerling op grond van zijn voorkennis in staat is. Bij het rekenen rond de tien kunnen tel-, rijg-, splits- en buursomstrategieën door het experimentele programma herkend worden.
In deze bijdrage zal aan de hand van precieze analyse van strategieën verduidelijkt worden, waarom een strategie in het algemeen alleen door een leerling ontwikkeld zal worden, als de daarvoor benodigde voorkennis aanwezig is.
De conclusie is: de wijze waarop verkortingen in rekenstrategieën plaatsvinden en de voorkeur voor niet-tel-strategieën, worden niet alleen door context en modelgebruik, maar ook door de beschikbare voorkennis beïnvloed. In de schoolpraktijk is het vanuit dit standpunt van belang aandacht te besteden aan precieze training van basisvaardigheden tot op een voldoende niveau ten behoeve van ontwikkeling van verdere inzichten en slimme strategieën van kinderen.

vier voorbeeldopgaven

De analyse in deze bijdrage zal gemaakt worden aan de hand van een viertal voorbeeldopgaven.

6 + 7
6 + 8
25 + 9
25 + 19

Deze vier zijn gekozen omdat ze passen bij het thema van deze conferentie. Het gaat in alle vier de opgaven om sprongen over de tien. Deze vier voorbeelden geven zoveel materiaal, dat ik afgezien heb van toevoeging van aftreksommen, die een nog veel grotere verscheidenheid aan mogelijke oplossingen bieden. Het in de inleiding genoemde programma Aritmeticus werkt niet alleen voor optellingen en aftrekkingen, maar is in staat strategieën te maken voor alle basisbewerkingen (cijferend en hoofdrekenend) tot duizend (in feite onbegrensd). Het is zó gemaakt dat het relatief gemakkelijk uit te breiden is voor bewerkingen met breuken en kommagetallen.

De vier bovenstaande voorbeeldopgaven zullen gevolgd worden in hun 'leergeschiedenis'. Niet echt, maar gesimuleerd. In opeenvolgende tabellen zullen oplossingen van Aritmeticus gegeven worden, die gemaakt zijn op basis van steeds meer voorkennis van een gefingeerde leerling, die verder Neo genoemd wordt. In dit artikel wordt het leren van Neo gesimuleerd, door zijn voorkennis te laten groeien. Dat kan door Aritmeticus eerder door hemzelf gevonden redeneringen aan het kennisbestand van Neo te laten toevoegen.

Het aardige is dat er zó met Aritmeticus simulatie van leerwegen mogelijk is. Later is het de bedoeling dat Aritmeticus 'echte' leerlingen gaat volgen en redeneringen van kinderen herkent en toevoegt aan het leerlingmodel, dat Aritmeticus van individuele leerlingen zal bijhouden. Dan zal Aritmeticus de kwaliteit van redeneringen kunnen beoordelen op grond van wat volgens het leerlingmodel voor de betreffende leerling bereikbaar is.

aspecten van een berekening

In figuur 1 worden enkele aandachtspunten genoemd, die verband houden met de 'kwaliteit van een berekening'.

- redeneerweg (gebruikte regels)
- gebruik van voorkennis (feiten, automatismes, routines)
- tijdsduur
- geheugenbelasting (tijdens de berekening te onthouden zaken)
- lengte van de berekening in aantal stappen
- complexiteit (aantal verschillende bewerkingen en relaties) van de tussenstadia
- representatiewisselingen
- probleem- deelprobleem wisselingen

figuur 1: kwaliteiten van een berekening.

gebruik van voorkennis

Gebruik van voorkennis speelt in de volgende analyse een belangrijke rol. In alle voorbeelden worden tellen, doortellen en het kunnen splitsen van getallen bekend verondersteld. Het splitsen van getallen behelst zowel het weten dat $45 = 40 + 5$ als het weten dat $40 + 5 = 45$. Ik ben geneigd deze laatste optelling niet als echte optelling te zien, maar als akoestische optelling:

vijf-en-veertig is hetzelfde als veertig-en-vijf.

Deze optelling is nauw verwant aan het kennen van de structuur van de telrij.

De kennis van de tafels van optelling en aftrekking zal gaandeweg de simulatie veranderd worden. Eerst zullen alleen de optellingen en aftrekkingen die onder de tien blijven (die je op je vingers kunt doen) bekend verondersteld worden, later zullen er meer toegelaten worden. Deze keus is erg kunstmatig. Als Aritmeticus later kinderen gaat volgen, dan wordt van elke individuele tafelsom onthouden hoe goed die bij de betreffende leerling beklijft. Dat kan Aritmeticus dan vervolgens gebruiken in verband met het testen of een opgave goed door een leerling gemaakt zal kunnen worden. Ook nu controleert Aritmeticus bij het genereren van oplossingen of de voor de redenering benodigde voorkennis aanwezig is. Als een leerling bijvoorbeeld $3 + 4$ nog niet weet, maar $3 + 5$ wel, dan zal Aritmeticus bij het generen van oplossingen voor $63 + 4$ een teloplossing of een buursomoplossing geven, terwijl hij voor $63 + 5$ ook de klassieke oplossing $63 + 5 = 60 + 8 = 68$ zal geven.

wat is splitsen?

Met het voorbeeld van $63 + 5$ is een belangrijk aspect van rekenstrategieën naar voren gekomen: rekent iemand die $63 + 5 = 68$ uitrekent wel op zo'n 'splitserige' manier? Is het niet eerder zó dat er iets gedacht wordt in de sfeer van:

drie en zestig plus vijf, drie en vijf is acht, achtenzestig , of:
drie-enzestig, vier-enzestig, ..., dus drie en vijf is acht-enzestig,
waarbij de 'enzestig' van drie-enzestig herhaald wordt.

Mijn conclusie is dat 63 + 5 = 68 op verscheidene manieren gedaan kan zijn. Het kan een verkorting van een telproces zijn, het kan een voortborduren op het hierboven genoemde splitsen van getallen zijn, en het kan een expliciete optelling zijn. Het kan ook een bijna synchrone vermenging van deze aanpakken zijn. Als echter een leerling expliciet zegt: 63 + 5 is 60 + 8, dan valt de telvariant mijns inziens af. De andere twee liggen zó dicht bij elkaar dat ze didactisch niet als zinvol te onderscheiden zijn.
Als nu in het verdere verloop van deze bijdrage optellingen van de vorm 60 + 3 voorkomen, gevolgd door het resultaat 63, dan wordt de hierboven bedoelde klankmatige optelling bedoeld. Als er een splitsing gemaakt wordt (67 = 60 + 7), dan gaat het niet om een 'harde' optelsplitsing, maar eerder om een akoestische splitsing.

afronden naar boven en buren

Als een leerling weet dat negen bijna tien is, dan kan 25 + 9 omgezet worden in 25 + 10 – 1. Een leerling denkt waarschijnlijk anders. Hij denkt eerder: 25 + 9 weet ik niet, maar 25 + 10 wèl en dan moet er straks nog één af. Aritmeticus noteert deze denkwijze in de formule 25 + 10 – 1. De denkwijze 'dit getal is bijna...' wordt in de tabellen aangeduid met afronden (naar boven). Nu is niet elke afronding naar boven zinvol. Het verschil moet te overzien zijn. Dat is het geval als het verschil een automatisme is en niet te groot. In de voorbeelden wordt het maximale verschil met 'maximum afstand voor afronden' aangeduid. In Aritmeticus is dit maximum leerling- en context-afhankelijk.
Iets vergelijkbaars gebeurt met buursommen: 6 + 7 kan worden herleid naar '5 + 5' en dan nog wat om het goed te maken. Ook hier geldt dat de verschillen gemakkelijk overbrugbaar moeten zijn.

welke oplossingen worden gemaakt?

Als Aritmeticus oplossingen maakt, dan genereert hij niet alle mogelijke oplossingen. Dat is niet nodig. Verschillen in volgorde binnen één formule worden niet onderscheiden. Als een leerling een andere volgorde heeft dan Aritmeticus, wordt de notatie van de leerling herkend en onthouden als modaliteit van een door Aritmeticus gevonden aanpak. Als er voor oplossing van een deelprobleem een feit, een automatisme of een routine beschikbaar is, dan wordt daaraan de voorkeur gegeven. Als een leerling in plaats van een automatisme toch een stapsgewijze uitwerking geeft van een aanpak die hij normaal als snel automatisme uit het hoofd afwerkt, dan wordt die uitwerking door Aritmeticus herkend, maar ook als regressie gekwalificeerd. Als een leerling een nieuwe oplossing produceert in plaats van een automatisme, dan zal ook dat door Aritmeticus herkend worden. Maar als iemand vraagt hoe een leerling een opgave vermoedelijk zal oplossen, dan geeft Aritmeticus de voorkeur aan feiten, automatismen en routines.
Ten slotte nog dit: Aritmeticus zoekt in principe naar korte oplossingen. Lange oplossingen kan hij ook vinden, maar deze voorbeelden worden tot korte oplossingen beperkt.

weergave van oplossingen in de tabellen

De oplossingen die in de tabellen gepresenteerd worden zijn stukken van de interne weergave van de oplossingen in het computerprogramma Aritmeticus. Het is uitdrukkelijk niet zo, dat deze oplossingen zó aan leerlingen gepresenteerd worden. Op dit moment wordt er gewerkt aan technieken om deze oplossingen in pijlentaal, met een getallenlijn, MAB of geld te kunnen laten zien aan de kinderen. De weg andersom werkt al een beetje: een leerling maakt een opgave in pijlentaal en zijn oplossing wordt geformaliseerd tot een rij formules. De interne representatie van die formules in Aritmeticus is veel complexer en bevat méér informatie dan de weergave in de tabellen. De tabellen zijn niet meer dan een medium om gemakkelijk te kunnen zien wat voor oplossingen Aritmeticus zoal kan maken. Sommige oplossingen komen in de tabellen eigenlijk niet goed tot hun recht. Ze zouden beter met materiaal of in een model weergegeven kunnen worden.

probleem-deelprobleem wisselingen

Als 34 + 5 wordt uitgerekend, dan wordt 34 gesplitst in 30 + 4. Vervolgens wordt 4 + 5 uitgerekend en wordt de dertig op de achtergrond gehouden. Als 4 + 5 uitgerekend is, dan wordt 'de dertig weer bij de negen gedacht': 39. Dat betekent dat de concentratie in deze berekening een paar keer verlegd wordt: Eerst naar de 34, dan naar 4 + 5 en ten slotte naar 30 + 9. De 4 + 5 kan als deelprobleem worden opgevat. Eigenlijk is dat een beetje zwaar woord, omdat er wellicht meer sprake is van een concentratie op 4 + 5 dan van een 'apart nemen van. Wiskundig gesproken kun je zeggen dat 4 + 5 een deelformule is van 30 + 4 + 5. In de opeenvolgende tabellen zal dit wisselen in concentratie tussen deelprobleem en hoofdprobleem ter sprake komen. In de tabel is een overgang naar een deelprobleem herkenbaar aan een inspringing:

```
34 + 5
30 + 4 + 5
        4 + 5
        9
30 + 9
39
```

Deze probleem-deelprobleem wisselingen kunnen gemakkelijk tot fouten aanleiding geven: er kan systeemscheiding optreden. Als een leerling bezig is met een deelprobleem kan hij de configuratie van het hoofdprobleem vergeten.

voorbeeld opgave 6 + 7

In tabel 1 wordt de opgave 6 + 7 uitgewerkt. Voor de gefingeerde leerling Neo, is 6 + 7 niet bekend. Aritmeticus is gevraagd een berekening maken. Als je Aritmeticus de vrije hand laat kunnen er tientallen(!) oplossingen gevonden worden. In dit geval geven we daarom wat beperkingen. We doen alsof Neo een oplossing gevonden heeft, waarbij hij opschreef:

```
 6 + 7
10 + 3
13
```

Een tweede voorwaarde is dat Aritmeticus niet al te lange redeneringen moet maken.
In principe doet het er niet toe of de oplossing van de leerling of van Aritmeticus in cijfers geschreven, bij een getallenlijn genoteerd of met het vijfrek gedaan wordt. Belangrijk is de structurering in tussenstappen.
De eerste oplossing is de gewone afronding naar tien. Neo beschikt over de benodigde voorkennis van 7 = 3 + 4. Een oplossing als 6 + 7 = 6 + 2 + 5 kan ook door Aritmeticus gevonden worden, maar is in dit geval niet relevant door de eis dat 10 + 3 als tussenstap moet voorkomen.
Daarnaast is 6 + 2 + 5 wellicht aardig, maar voldoet het niet erg aan een norm van generaliseerbaarheid en effectiviteit. 6 + 2 + 5 is niet effectief, omdat de opgave 8 + 5 niet bekend of automatisch is, en eigenlijk eenzelfde soort oplossing vereist als 6 + 7. Aardig is dat Aritmeticus ook teloplossingen geeft. Niet voor 6 + 4, want dat is een feit, en ook niet voor 6 + 7 zelf, omdat dan de tussenstap 10 + 3 niet voorkomt, maar wel voor 10 + 3 (tabel 1, kolom 1, oplossing 2). De teloplossing 3 + 10 (tabel 1, kolom 2, oplossing 3) is

ineffectief, maar wordt volledigheidshalve toch gegeven. In de tweede kolom van tabel 1 worden vergelijkbare oplossingen voor 6 + 7 gegeven.

tabel 1

Aanvullen en afronden van 6 + 7 met 10 + 3 als tussenstap
'→' duidt formule aan die verplicht moet voorkomen
Voorkennis:
- tellen
- splitsen van getallen
- optellingen (a + b = c) en aftrekkingen (a – b = c) onder de tien
  (a < = 10, b < = 10, c < = 10)
- maximum afstand voor afronden: twee

→ 6 + 7 6 + 4 + 3 6 + 4 10 → 10 + 3 → 13	→ 6 + 7 7 + 3 + 3 7 + 3 10 → 10 + 3 → 13		
→ 6 + 7 6 + 4 + 3 6 + 4 10 → 10 + 3 10 11 12 → 13	→ 6 + 7 7 + 3 + 3 7 + 3 10 → 10 + 3 10 11 12 → 13		
→ 6 + 7 6 + 4 + 3 6 + 4 10 → 10 + 3 3 4 5 6 7 8 9 10 11 12 → 13	→ 6 + 7 7 + 3 + 3 7 + 3 10 → 10 + 3 3 4 5 6 7 8 9 10 11 12 → 13		

Als tabel 1 wordt bekeken met de eerdergenoemde kwaliteiten van berekeningen (fig.1), dan valt op dat de kortste oplossingen de meeste inzet van voorkennis vergen. De langste oplossingen vergen de minste voorkennis, maar kunnen door hun lengte gemakkelijk tot (tel-)fouten leiden. Op grond van die ervaring zetten veel kinderen liever de tien voorop dan de drie.

In tabel 2 worden buursomoplossingen gegeven. Daarbij wordt 6 + 7 herleid tot 4 + 6 en 5 + 5. Er wordt geen herleiding tot 6 + 4 gemaakt, want dat is een aanvulling tot de tien, die in tabel 1 al aan de orde kwam. 4 + 6 en 5 + 5 zijn allebei 10. Geen wonder want Aritmeticus moet bij 10 + 3 uitkomen. Het verschil tussen beide eerste kolommen is alleen een volgorde-verschil: wordt eerst de compensatie gedaan of wordt eerst de buur gedaan? Merk op dat 4 + 6 + 2 + 1 niet door Aritmeticus uitgewerkt wordt tot 10 + 2 + 1 = 12 + 1 = 13 omdat 10 + 3 als tussenresultaat verwacht wordt.

Merk op dat 6 + 6 niet als buur gebruikt wordt omdat 6 + 6 niet bekend is als feit en omdat 6 + 6 niet tot 10 + 3 leidt.

De buur 5 + 5 past goed bij het rekenen met het vijfrek.

Belangrijk is om in te zien dat de gegeven buursomoplossingen steeds twee probleem-deelprobleem wisselingen laten zien. Deze soort buren zijn daarom alleen veilig als een tabel 2

Buursommen van 6 + 7 met 10 + 3 als tussenstap
'→' duidt formule aan die verplicht moet voorkomen
Voorkennis:
- tellen
- splitsen van getallen
- optellingen (a + b = c) en aftrekkingen (a – b = c) onder de tien
  (a < = 10, b < = 10, c < = 10)
- maximum afstand voor buren: twee

→ 6 + 7 4 + 6 + 2 + 1 4 + 6 10 10 + 2 + 1 2 + 1 3 → 10 + 3 → 13	→ 6 + 7 4 + 6 + 2 + 1 2 + 1 3 4 + 6 + 3 4 + 6 10 → 10 + 3 → 13	→ 6 + 7 5 + 5 + 1 + 2 5 + 5 10 10 + 1 + 2 1 + 2 3 → 10 + 3 → 13	→ 6 + 7 5 + 5 + 1 + 2 1 + 2 3 5 + 5 + 3 5 + 5 10 → 10 + 3 → 13
→ 6 + 7 4 + 6 + 2 + 1 4 + 6 10 10 + 2 + 1 2 + 1 3 → 10 + 3 11 12 → 13	→ 6 + 7 4 + 6 + 2 + 1 2 + 1 3 4 + 6 + 3 4 + 6 10 → 10 + 3 10 11 12 → 13	→ 6 + 7 5 + 5 + 1 + 2 5 + 5 10 10 + 1 + 2 1 + 2 3 → 10 + 3 10 11 12 → 13	→ 6 + 7 5 + 5 + 1 + 2 1 + 2 3 5 + 5 + 3 5 + 5 10 → 10 + 3 10 11 12 → 13
→ 6 + 7 4 + 6 + 2 + 1 4 + 6 10 10 + 2 + 1 2 + 1 3 → 10 + 3 3 4 5 6 7 8 9 10 11 12 → 13	→ 6 + 7 4 + 6 + 2 + 1 2 + 1 3 4 + 6 + 3 4 + 6 10 → 10 + 3 3 4 5 6 7 8 9 10 11 12 → 13	→ 6 + 7 5 + 5 + 1 + 2 5 + 5 10 10 + 1 + 2 1 + 2 3 → 10 + 3 3 4 5 6 7 8 9 10 11 12 → 13	→ 6 + 7 5 + 5 + 1 + 2 1 + 2 3 5 + 5 + 3 5 + 5 10 → 10 + 3 3 4 5 6 7 8 9 10 11 12 → 13

leerling de betreffende deelbewerkingen drommels goed weet. Anders heb je goed kans dat hij door de dubbele probleem-deelprobleem wisseling in systeemscheiding, twijfel en verwarring raakt.
Tegen deze achtergrond is een strategie als $6 + 7 = 6 + 6 + 1 = 12 + 1$ veel aardiger, omdat hier maar één deelprobleem wisseling plaats heeft en de optelling met 1 niet problematisch is. Maar zoals gezegd is deze oplossing niet terzake omdat Neo $10 + 3$ als tussenstap gebruikte.

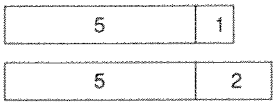
gebruik van modellen

Dit laatste voorbeeld van de probleem-deelprobleem wisselingen kan worden ondervangen door opschrijven, visualiseren of materialiseren:

genoteerd:

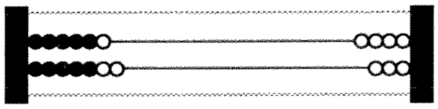
6 + 7 = 5 + 5 + 1 + 2

en getekend (fig.2).



figuur 2

en gematerialiseerd (fig.3).



figuur 3

Deze weergaven helpen overzicht te houden over wat je aan het doen bent. Maar zo gauw je dezelfde handeling uit het hoofd probeert te doen, moet je mentaal overzicht houden over wat je aan het doen bent. Zolang je materiaal gebruikt of tekent, is het niet zo erg als je deelbewerkingen niet zo gemakkelijk kunt maken. Maar als je dat materiaal niet

tabel 3

Aanvullen en afronden van 6 + 8 met 10 + 4 als tussenstap
'→' duidt formule aan die verplicht moet voorkomen
Voorkennis:
- tellen
- splitsen van getallen
- optellingen (a + b = c) en aftrekkingen (a – b = c) onder de tien (a < = 10, b < = 10, c < = 10)
- maximum afstand voor afronden: twee

→ 6 + 8 6 + 4 + 4 6 + 4 10 → 10 + 4 → 14	→ 6 + 8 8 + 2 + 4 8 + 2 10 → 10 + 4 → 14	→ 6 + 8 6 + 10 – 2 6 – 2 4 → 4 + 10 → 14
→ 6 + 8 6 + 4 + 4 6 + 4 10 → 10 + 4 10 11 12 13 → 14	→ 6 + 8 8 + 2 + 4 8 + 2 10 → 10 + 4 10 11 12 13 → 14	→ 6 + 8 6 + 10 – 2 6 – 2 4 → 4 + 10 10 11 12 13 → 14
→ 6 + 8 6 + 4 + 4 6 + 4 10 → 10 + 4 4 5 6 7 8 9 10 11 12 13 → 14	→ 6 + 8 8 + 2 + 4 8 + 2 10 → 10 + 4 4 5 6 7 8 9 10 11 12 13 → 14	→ 6 + 8 6 + 10 – 2 6 – 2 4 → 4 + 10 4 5 6 7 8 9 10 11 12 13 → 14

meer hebt, dan kun je in de moeilijkheden komen, doordat de deelbewerking zoveel concentratie vergt dat er systeemscheiding gaat optreden en het overzicht verloren gaat. Dit is bepaald geen pleidooi tegen materiaal of modelgebruik. Maar een belangrijke didactische conclusie is wel dat er problemen op kunnen treden als kinderen van materiaal of model over (moeten) gaan naar mentaal rekenen. Dan wordt kennis van deelvaardigheden van groot belang.

Een van de doelen van het ISWA-project is nu juist het oefenen van deelvaardigheden te waarborgen tijdens het werken met modellen. Vandaar dat in de paragraaf 'Welke oplossingen worden gemaakt?', ter sprake kwam dat Aritmeticus regressie in deeloplossingen, waarvoor een automatisme of feit bestaat, moet kunnen herkennen.
De teloplossingen in de tweede rij van de tabellen 1 en 2 kunnen in de psychologie van Neo opgevat worden als gevolg van onzekerheid ten gevolge van de terugkeer uit het deelprobleem 6 + 4 of 8 + 2. Bijgevolg vertoont Neo de wat zwakkere teloplossing in plaats van de akoestische optelling. In een dergelijk geval kan Aritmeticus Neo er op wijzen dat Neo best in staat is om 10 + 4 = 14 direct zonder tellen te doen.

voorbeeld opgave 6 + 8

Deze opgave geeft bijna dezelfde resultaten als 6 + 7. Het enige nieuwe is de derde kolom van tabel 3. Daar wordt 8 gezien als 'in de buurt van tien.' Aritmeticus geeft hier alleen de uitwerking via 6 + 10 - 2 en vervolgens 6 – 2 = 4, omdat uitwerking via 16 – 2 niet leidt tot de vereiste tussenstap 10 + 4. Belangrijk is, dat Neo volgens Aritmeticus 6 – 2 weet, en 16 – 2 tellend, of via splitsing van zestien zou moeten doen.

tabel 4

Aanvullen en afronden van 25 + 9 Voorkennis: – tellen – splitsen van getallen – optellingen (a + b = c) en aftrekkingen (a – b = c) onder de tien (a < = 10, b < = 10, c < = 10) – tellen in tientallen – maximum afstand voor afronden: twee					
25 + 9 20 + 5 + 10 – 1 20 + 10 30 30 + 5 – 1 30 + 5 35 35 – 1 34	25 + 20 + 5 + 10 – 1 20 + 10 30 30 + 5 – 1 5 – 1 4 30 + 4 34	25 + 9 20 + 5 + 10 – 1 5 – 1 4 20 + 4 + 10 20 + 10 30 30 + 4 34	25 + 9 25 26 27 28 29 30 31 32 33 34	25 + 9 20 + 5 + 4 25 + 5 25 26 27 28 29 30 30 + 4 34	25 + 9 20 + 5 + 9 20 + 9 29 29 + 5 29 30 31 32 33 34

tabel 5

Bijgeleerde voorkennis typen als: 23 + 5, 27 – 5, 25 + 5, 34 – 4					
25 + 9 20 + 5 + 4 25 + 5 30 30 + 4 34	25 + 9 20 + 5 + 4 25 + 5 30 30 + 4 30 31 32 33 34				

voorbeeld opgave 25 + 9

Bij deze opgave worden wat langere redeneringen toegelaten. In tabel 4 is Aritmeticus met weinig voorkennis aan het werk geweest. Opvallend is dat er alleen teloplossingen zijn en oplossingen met afronding naar boven van de 9 en splitsing van de 25. De eerste drie oplossingen zijn vrij slim, maar slim uit pure armoede. Neo beschikt behalve tellen niet over routines om 'vanaf de 25' iets te kunnen doen. Zelfs een splitsing in de richting van 25 + 9 = 20 + 5 + 9 en dan naar 5 + 9 helpt niet omdat Neo 5 + 9 niet weet. Neo zou 5 + 9 ook weer moeten tellen of via een splitsing moeten uitrekenen:

```
25 +  9
20 +  5 +  9
           5 +  9
                5 + 5
              10
          10 +  4
          14
20 + 14
```

Met deze 20 + 14 zou Neo weer opnieuw in de problemen raken. Opvallend in deze berekening is dat 5 + 5 'twee diep' staat: er is twee keer een deelprobleem apart genomen. In termen van figuur 1: de redeneerweg wordt lang en er ontstaat een grote geheugenbelasting, doordat onthouden moet worden dat er 'later' nog twintig en nog vier bij moeten. De geheugenbelasting en de complexiteit van de stappen wordt groot.
Van pure ellende moet Aritmeticus wel naar 20 + 10 toewerken (tabel 4), want tellen -ook in sprongen van tien - kan Neo goed. Een complicerende factor is dat door de afronding naar boven onthouden moet worden dat de één van 10 – 1 afgetrokken moet worden. Als je alleen met optellen te maken hebt hoef je alleen maar getallen te onthouden, en geen bewerkingen ertussen. Heb je een mengsel van optellingen en aftrekkingen, dan moet je bij elk getal onthouden wat ermee gedaan moet worden. De laatste oplossing van tabel 4 is erg aardig en uit de praktijk goed bekend: de tellende leerling zet het grootste getal voorop: 25 + 9 = 29 + 5, en telt dan verder.

Ik heb Neo opgaven binnen een tiental en tot op het tiental laten bijleren:

```
23 + 5
20 + 3 + 5
     3 + 5
     8
20 + 8
28
```

en dergelijke zijn nu vertrouwd. In feite beschikt Neo nu over de kennis die ik in de paragraaf 'Wat is splitsen?' aangegeven heb. Er ontstaan nu interessante rijg-oplossingen, zoals die in het Speerpunt rekenen beschreven zijn. De 25 hoeft niet meer gesplitst te worden (tabel 5). Oplossing 1 van tabel 5 kan als verkorting van oplossing 5 van tabel 4 gezien worden. De routine voor 25 + 5 maakte deze verkorting mogelijk. Ook dit is een belangrijk punt van Aritmeticus: dit soort verkortingen kunnen worden geconstateerd en gevolgd. Als een leerling in een dergelijk geval bij 25 + 5 van het automatisme terugvalt naar een teloplossing, dan zal Aritmeticus dat aan de leerling direct terug kunnen melden.
Tellen in de zuivere tientallen: dertig, veertig, vijftig, ..., is anders dan het tellen in sprongen van tien. Tellen in sprongen van tien heeft met akoestische splitsing te maken. Je gebruikt dan dat je in de tientallen kunt tellen met behoud van de eenheden. Dit in tegenstelling tot tellen in de tientallen, wat in feite alleen neerkomt op analogie aan tellen onder de tien met aandacht voor de uitgangen '-tig' en wat fonetische aanpassing. Tellen

in sprongen van tien moet apart geleerd worden. Dat kan onder andere door te tellen, door gebruik van MAB-materiaal of getallenlijn maar ook door redeneringen als:

23 + 30
20 + 3 + 30
20 + 30
20
30
40
50
50 + 3
53

Deze aanpak komt erop neer dat je telt in de tientallen en steeds de drie op de achtergrond houdt en toevoegt.
Deze voorkennis maakt voor Neo weer nieuwe oplossingen mogelijk (tabel6). Het afronden van negen naar tien is nu een heel interessante optie geworden. De eerste oplossing in tabel 6 is van het type eentje meer, eentje minder. Bij de tweede en derde gaat het echt om afronding naar boven. Als dan ten slotte alle tafels gekend zijn wordt voor Neo de oplossing via 5 + 9 mogelijk (tabel 7). Dankzij het feit dat Neo inmiddels kan tellen en optellen in de tientallen is de stap 20 + 14 voor hem gemakkelijk.

tabel 6

Bijgeleerde voorkennis typen als: 23 + 10						
25 + 9 9 + 1 + 24 9 + 1 10 10 + 24 34	25 + 9 25 + 10 − 1 25 + 10 35 35 − 1 34	25 + 9 25 + 10 − 1 25 − 1 24 24 + 10 34				

tabel 7

Bijgeleerde voorkennis alle sommen met operanden onder de tien						
25 + 9 20 + 5 + 9 5 + 9 14 20 + 14 34						

leerwegen

Ik wijs er met nadruk op dat de zo gesimuleerde leerweg er heel anders uit zou zien als Neo eerst alle tafels zou leren. Het is de bedoeling van het ISWA-project dat kinderen in al dit soort onderling verschillende leerwegen gevolgd en gecoached kunnen worden. Dat betekent dat de waardering voor oplossingen steeds afhankelijk is van het reeds bereikte niveau van de leerling. Aritmeticus zal later commentaar op een berekening kunnen geven. Hij kan kinderen wijzen op regressie of juist een compliment geven om nieuwe vondsten of vooruitgang in routine en feitenkennis.

voorbeeld opgave 25 + 19

Met dit voorbeeld stel ik uw geduld op de proef. Maar het is een voorbeeld dat illustreert in wat voor ellende een leerling belandt, die te weinig voorkennis heeft. In tabel 8 zijn 21 verschillende oplossingen gegeven voor 25 + 19, waarbij ervan uitgegaan wordt dat Neo weinig voorkennis heeft.

tabel 8

<table>
<tr><td colspan="7">Aanvullen en afronden van 25 + 19
Voorkennis:
– tellen
– splitsen van getallen
– optellingen (a + b = c) en aftrekkingen (a – b = c) onder de tien
(a < = 10, b < = 10, c < = 20)
– maximum afstand voor afronden: twee</td></tr>
<tr>
<td>25 + 19
20 + 5 + 20 – 1
20 + 20
2 + 2
4
40
40 + 5 – 1
40 + 5
45
45 – 1
44</td>
<td>25 + 19
20 + 5 + 20 – 1
20 + 20
2 + 2
4
40
40 + 5 – 1
5 – 1
4
40 + 4
44</td>
<td>25 + 19
20 + 5 + 20 – 1
5 – 1
4
20 + 4 + 20
20 + 20
2 + 2
4
40
40 + 4
44</td>
<td>25 + 19
20 + 5 + 20 – 1
20 + 20
2 + 2
4
40
40 + 5 – 1
5 – 1
4
40 + 4
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 20 – 1
20 + 1
4
20 + 4 + 20
20 + 20
2 + 2
4
40
40 + 4
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 10
2 + 1
3
30
30 + 5 + 9
30 + 9
39
39 + 5
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 20 – 1
20 + 20
2 + 2
4
40
40 + 5 – 1
40 + 5
40
41
42
43
44
45
45 – 1
44</td>
</tr>
<tr>
<td>25 + 19
20 + 5 + 20 – 1
20 + 20
2 + 2
4
40
40 + 5 – 1
40 – 1
39
39 + 5
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 10
2 + 1
3
30
30 + 5 + 9
30 + 5
35
35 + 9
35
36
37
38
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 10
20
21
22
23
24
25
26
27
28
29
30
30 + 5 + 9
30 + 9
39
39 + 5
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 9
29
29 + 5 + 10
29 + 5
29
30
31
32
33
34
34 + 10
34
35
36
37
38
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 9
29
29 + 5 + 10
29 + 10
29
30
31
32
33
34
35
36
37
38
39
34 + 5
39
40
41
42
43
44</td>
<td>25 + 19
25 + 10 + 9
25 + 10
25
26
27
28
29
30
31
32
33
34
35
35 + 9
35
36
37
38
39
40
41
42
43
44</td>
<td>25 + 19
25 + 10 + 9
25 + 9
25
26
27
28
29
30
31
32
33
34
34 + 10
34
35
36
37
38
39
40
41
42
43
44</td>
</tr>
<tr>
<td>25 + 19
20 + 5 + 10 + 9
20 + 10
2 + 1
3
30
30 + 5 + 9
30 + 5
30
31
32
33
34
35
35 + 9
35
36
37
38
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 10
2 + 1
3
30
30 + 5 + 9
30 + 9
30
31
32
33
34
35
36
37
38
39
39 + 5
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 5
25
25 + 10 + 9
25 + 10
25
26
27
28
29
30
31
32
33
34
35
35 + 9
35
36
37
38
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 5
25
25 + 10 + 9
25 + 9
25
26
27
28
29
30
31
32
33
34
34 + 10
34
35
36
37
38
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 10
20
21
22
23
24
25
26
27
28
29
30
30 + 5 + 9
30 + 5
35
35 + 9
35
36
37
38
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 5
20
21
22
23
24
25
25 + 10 + 9
25 + 9
25
26
27
28
29
30
31
32
33
34
34 + 10
34
35
36
37
38
39
40
41
42
43
44</td>
<td>25 + 19
20 + 5 + 10 + 9
20 + 9
20
21
22
23
24
25
26
27
28
29
29 + 5 + 10
29 + 5
29
30
31
32
33
34
34 + 10
34
35
36
37
38
39
40
41
42
43
44</td>
</tr>
</table>

Zelfs tellen in de tientallen kan Neo niet. Duidelijk is dat de opgave 25 + 19 een formidabele taak is.
In de eerste rij van tabel 8 komen geen omvangrijke tellingen voor. Net als bij 25 + 9 wordt uit pure nood gebruik gemaakt van splitsingen en afrondingen naar boven. 20 + 20 is dan het lastigste deelprobleem. Om dat uit te rekenen moet Aritmeticus teruggaan naar zoiets als 'twee staafjes van tien (tientallen) en nog eens twee'.
Dat heeft tot gevolg dat in de uitwerking van 25 + 19 oplossingen voorkomen, waar weer twee maal achtereen een deelprobleem apart genomen moet worden, met alle gevaren van systeemscheiding van dien. Alleen de zevende oplossing van de eerste rij is anders: Daar wordt negentien gesplitst en wordt 20 + 10 gedaan. In termen van figuur 1 moeten al deze oplossingen als 'zwaar' gekwalificeerd worden. Veel getallen, vaak complexe structuren, steeds twee probleem-deelprobleem wisselingen, waarvan er vaak één twee stappen diep gaat.

tabel 9

Bijgeleerde voorkennis typen als: 30 + 10						
25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 40 + 5 45 45 − 1 44	25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 5 − 1 4 40 + 4 44	25 + 19 20 + 5 + 20 − 1 5 − 1 4 20 + 4 + 20 20 + 20 40 40 + 4 44	25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 5 − 1 4 40 + 4 40 41 42 43 44	25 + 19 20 + 5 + 20 − 1 5 − 1 4 20 + 4 + 20 20 + 20 40 40 + 4 40 41 42 43 44	25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 30 + 9 39 39 + 5 39 40 41 42 43 44	25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 40 + 5 40 41 42 43 44 45 45 − 1 44
25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 40 − 1 39 39 + 5 39 40 41 42 43 44	25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 30 + 5 35 35 + 9 35 36 37 38 39 40 41 42 43 44	25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 30 + 5 30 31 32 33 34 35 35 + 9 35 36 37 38 39 40 41 42 43 44	25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 30 + 9 30 31 32 33 34 35 36 37 38 39 39 + 5 39 40 41 42 43 44			

tabel 10

Bijgeleerde voorkennis typen als: 34 + 5, 34 + 6, 45 –5, 45– 2, 56 + 8						
25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 30 + 5 35 35 + 9 44	25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 30 + 9 39 39 + 5 44	25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 40 + 5 45 45 − 1 44	25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 40 − 1 39 39 + 5 44			

modellen

Ook bij deze opgave kan gebruik van modellen veel verzachten. Bijvoorbeeld met een getallenlijn, met MAB-materiaal of met een abacus is het mogelijk overzicht te houden. Tekening of materiaal doen dan dienst als een uitwendig geheugen. Bovendien maakt modelgebruik het mogelijk dat je naar de stand van de berekening op papier kunt kijken. Je kunt dan energie vrijmaken om te reflecteren op je aanpak. Als een leerling met te weinig voorkennis te snel sommen uit het hoofd moet doen, dan wordt hij gedwongen te gaan tellen of grote en gecompliceerde splitsingen te gaan maken.

tabel 11

Bijgeleerde voorkennis typen als: 23 + 10, 20 + 14						
25 + 19 25 + 5 + 14 25 + 5 30 30 + 14 44	25 + 19 25 + 10 + 9 25 + 10 35 35 + 9 44	25 + 19 25 + 10 + 9 25 + 9 34 34 + 10 44	25 + 19 25 + 5 + 19 20 + 19 39 + 5 39 + 5 44	25 + 19 19 + 1 + 24 19 + 1 20 20 + 24 24 34 44	25 + 19 20 + 5 + 19 5 + 19 24 20 + 24 24 34 44	25 + 19 25 + 20 − 1 25 + 20 25 35 45 45 − 1 44
25 + 19 25 + 20 − 1 25 − 1 24 24 + 20 24 34 44	25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 30 + 5 35 35 + 9 44	25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 30 + 9 39 39 + 5 44	25 + 19 20 + 5 + 10 + 9 20 + 10 30 30 + 5 + 9 5 + 9 14 30 + 14 44	25 + 19 20 + 5 + 10 + 9 20 + 9 29 29 + 5 + 10 29 + 5 34 34 + 10 44	25 + 19 20 + 5 + 10 + 9 20 + 9 29 29 + 5 + 10 29 + 10 39 39 + 5 44	25 + 19 20 + 5 + 10 + 9 5+ 10 15 20 + 15 + 9 20 + 15 35 35 + 9 44
25 + 19 20 + 5 + 10 + 9 5 + 9 14 20 + 14 + 10 20 + 14 34 34 + 10 44	25 + 19 20 + 5 + 10 + 9 5 + 9 14 20 + 14 + 10 20 + 10 30 30 + 14 44	25 + 19 20 + 5 + 10 + 9 10 + 9 19 20 + 5 + 19 20 + 19 39 39 + 5 44	25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 40 + 5 45 45 − 1 44	25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 40 − 1 39 39 + 5 44	25 + 19 20 + 5 + 20 − 1 20 + 20 40 40 + 5 − 1 5 − 1 4 40 + 4 44	25 + 19 20 + 5 + 20 − 1 5 − 1 4 20 + 4 + 20 20 + 20 40 40 + 4 44
25 + 19 20 + 5 + 10 + 9 5 + 10 15 20 + 15 + 9 15 + 9 24 20 + 24 24 34 44	25 + 19 20 + 5 + 10 + 9 10 + 9 19 20 + 5 + 19 5 + 19 24 20 + 24 24 34 44	25 + 19 20 + 5 + 20 − 1 20 + 5 25 25 + 20 − 1 25 + 20 25 35 45 45 − 1 44	25 + 19 20 + 5 + 20 − 1 20 + 5 25 25 + 20 − 1 25 − 1 24 24 + 20 24 34 44	25 + 19 20 + 5 + 20 − 1 20 − 1 19 19 + 5 + 20 19 + 5 24 24 + 20 24 34 44	25 + 19 20 + 5 + 20 − 1 5 + 20 25 20 + 25 − 1 20 + 25 25 35 45 45 − 1 44	25 + 19 20 + 5 + 20 − 1 5 + 20 25 20 + 25 − 1 25 − 1 24 20 + 24 24 34 44
25 + 19 20 + 5 + 20 − 1 5 − 1 4 20 + 4 + 20 20 + 4 24 + 20 24 34 44	25 + 19 20 + 5 + 20 − 1 5 − 1 4 20 + 4 + 20 4 + 20 20 + 24 24 34 44					

Ik meen ervoor te moeten waarschuwen dat gebruik van modellen niet vanzelf tot verbetering van benodigde voorkennis leidt. Evenmin leidt het niet vanzelf tot goede geheugenoefening, in verband met het onthouden van complexe situaties en het weer ophalen van feitenkennis, terwijl het werkgeheugen belast is. Bij de overgang van materiaal en getekende modellen naar mentaal rekenen kan daarom een regressie in de kwaliteit van de redeneringen van een leerling optreden. Wiskobas heeft deze problematiek ondervangen door geleidelijke (progressieve) schematisering van modelgebruik te bevorderen.

Deze geleidelijke schematisering kan goed bevorderd worden door ook aandacht voor benodigde voorkennis te hebben, in de vorm van aparte, op snelheid en accuratesse gerichte oefening. Deze oefening kan expliciet en apart gedaan worden, maar ook impliciet door tijdens het werken met modellen aandacht te hebben voor de snelheid waarmee een leerling deelproblemen oplost. In het ISWA-project worden beide beoogd.

verkorting van redeneringen door toename van voorkennis

In tabel 9 zijn oplossingen weergegeven, die mogelijk worden als Neo in de tientallen kan tellen. Er verschijnen nu uit de schoolpraktijk vertrouwde oplossingen. Er ligt nog steeds grote nadruk op oplossingen waarbij negentien afgerond wordt naar twintig.
Pas in tabel 10, als Neo optellingen in en over het tiental heeft bijgeleerd, komt daar verandering in. Een voorbeeld van het leerproces dat hier plaatsvindt, is eerder in detail besproken in het voorbeeld van 25 + 9.

verborgen lacunes in voorkennis

In tabel 11 worden vervolgens vrij snelle korte oplossingen beschreven, waarin geregen wordt. Deze oplossingen zijn gebruikelijk in de schoolpraktijk. Eén soort oplossingen ontbreekt nog. Dat zijn oplossingen met de stap 9 + 5 = 14. Want de tafels over de tien heen kent Neo nog steeds niet. Dat is iets dat in de schoolpraktijk ook gemakkelijk kan gebeuren. Het lijkt erop dat een leerling aardig en efficiënt kan rekenen, maar onder die handigheid verbergt hij gebrekkige basiskennis. In het ISWA-project gaat het er om dit soort lacunes, die gemakkelijk aan de aandacht kunnen ontsnappen te voorkomen, door nauwgezette toetsing van impliciet en expliciet gebruikte voorkennis.

slot

De gegeven voorbeelden stroken met de ervaring dat juist zwakke leerlingen neiging hebben ongewone rekenstrategieën te bedenken. Ze stroken ook met de oude onderwijservaring dat voldoende voorkennis wenselijk is om goede rekengewoonten te bevorderen. Ze passen ook bij het Wiskobasinzicht dat geleidelijke (progressieve) schematisering vanuit modellen naar mentaal rekenen vooral voor zwakke leerlingen van belang kan zijn, omdat ze dan complexere aan rekentaken in de groep mee kunnen doen (differentiatie) en tegelijk meer elementaire vaardigheden kunnen oefenen. De waarschuwing daarbij is wel dat er heel precies bijgehouden moet worden of die meer elementaire vaardigheden ook inderdaad groeien en op een gewenst niveau komen. Dit punt is cruciaal, want als die elementaire vaardigheden niet meegroeien gaat dat gebrek negatieve invloed hebben op de kwaliteit van oplossingen. Speciaal geldt dat leerlingen met te weinig voorkennis gemakkelijk splitsers worden in plaats van rijgers. En we willen juist zo graag dat kinderen rijgers worden, omdat rijgstrategieën in termen van figuur 1 te prefereren zijn: minder complexiteit en minder lange berekeningen met als prijs: meer voorkennis. Deze problematiek rond geleidelijk schematiseren is een van de kernen van het ISWA-project van de SLO. Het leerplanontwikkelingsprobleem is: hoe moet een sterk gedifferentieerd oefen-curriculum, passend bij realistische en andere methodes er uit zien? Daarin speelt enerzijds ruimte voor eigen vondsten een belangrijke rol, en aan de andere kant is zorgvuldige evaluatie van vaardigheden en redeneringen van belang. Dat is in praktijk van de school een complexe zaak. Vandaar dat de SLO expertise wil opbouwen inzake computergebruik als planningsinstrument, toetsinstrument voor de leraar en leeromgeving voor de leerlingen. Het ISWA-project leidt in de richting van een dynamisch, - door de leerweg van de leerling en door de didactische keuzen van de leraar bepaald - curriculum, dat in de vorm van een computersysteem door de leraar gebruikt kan worden in sterk heterogene groepen.