VERVOLGVEL NR.: DATUM:

EEN ANALYSE VAN HET SPEL "MEMORY"

Een verslag van de 65-middagenopdracht uitgevoerd in de periode januari / juni 1983 bij de vakgroep 'Informatica'.

BEGELEIDER: Dr. F. Göbel

DOOR: Sabih H. Gerez (EL)

VERSLAG / RAPPORT PROEFNR. NAAM:

VERVOLGVEL NR.: i DATUM:

SAMENVATTING

Er wordt uitgegaan van een model van het spel Memory met perfect geheugen. M.b.v. dynamisch programmeren kan voor elke toestand in het spel de optimale taktiek bepaald worden.
Een groot deel van het verslag is gewijd aan de theoretische onderbouwing van de resultaten, die d.m.v. een computerprogramma zijn verkregen. Het is slechts gelukt om een deel van de regelmaat in deze resultaten te verklaren.

VERSLAG / RAPPORT PROEFNR. NAAM:

VERVOLGVEL NR. ii DATUM:

. . .

"Living backwards!" Alice repeated in great astonishment. "I never heard of such a thing!".

"– but there's one great advantage in it, that one's memory works both ways."

"I'm sure mine only works one way," Alice remarked. "I can't remember things before they happen."

"It's a poor sort of memory that only works backwards," the Queen remarked.

. . .

Uit: Lewis Carroll –
Through the Looking-Glass

VERSLAG / RAPPORT PROEFNR. NAAM:

VERVOLGVEL NR. iii DATUM:

INHOUDSOPGAVE

(vervolg inhoudsopgave)

Blz

HOOFDSTUK 1: OPDRACHTSOMSCHRIJVING EN INLEIDING

De opdrachtsomschrijving die mij voor deze opdracht deed kiezen, was de volgende:

"Gevraagd wordt het spel Memory te modelleren, zodat een optimale strategie m.b.v. dynamisch programmeren kan worden bepaald. Een computerprogramma dient de strategieën numeriek te bepalen, met rationele aritmetiek. Deze uitkomsten dienen als basis voor een meer theoretische aanpak. Afhankelijk van de resultaten daarvan kunnen nog bepaalde generalisaties van het spel worden geanalyseerd"

In dit verslag kan men lezen hoe ik deze opdracht heb uitgewerkt.
Hoofdstuk 2 geeft een beschrijving van het spel (voor degene die het spel nog niet kent) en legt vervolgens uit welke aspecten van het spel in het model zullen worden meegenomen.

Hoofdstuk 3 werkt het model uit zodat m.b.v. dynamisch programmeren de optimale strategie kan worden bepaald. Vervolgens komen daar de resultaten van de berekening en de theoretische onderbouwing ervan aan de orde. Het zal blijken dat het model zal moeten worden aangepast voordat de theoretische onderbouwing verder kan worden gevoerd.

De resultaten en de theoretische onderbouwing van het aangepaste, vereenvoudigde model komen dan in Hoofdstuk 4 aan de orde.

Hoofdstuk 5 geeft een aantal mogelijkheden aan voor variaties op het model van het spel. Wegens tijdgebrek zijn die variaties niet of nauwelijks onderzocht.

In dit verslag zal men weinig verwijzingen vinden naar de computerprogramma's die geschreven zijn. Het programmeren zelf was, volgens mij, van ondergeschikt belang. Ik volsta dan ook met het aangeven van de manier van berekening en het presenteren van de resultaten. De programmateksten, die in een bijlage zijn opgenomen, zijn hopelijk van voldoende commentaar voorzien om begrijpelijk te zijn.

Wat betreft de rationale aritmetiek, al in een vroeg stadium bleek dat de voordelen ervan waren overschat, zodat ze in de uiteindelijke programma's en resultaten niet is terug te vinden.

Een literatuurlijst ontbreekt in dit verslag. Van de literatuur die ik heb bestudeerd, heb ik alleen nuttig gebruik kunnen maken van het hoofdstuk over dynamisch programmeren in:
B.T. Houlden (ed.), Some Techniques in Operational Research, English Universities Press Ltd., London 1962.

Ik zal in het vervolg van dit verslag vaak de "we"-vorm gebruiken. Het "we" bedoel ik meestal mijzelf, maar soms ook mijn begeleider, de heer Göbel. Zijn suggesties zijn vaak van doorslaggevende betekenis geweest voor de voortgang van de opdracht.

HOOFDSTUK 2: BESCHRIJVING EN MODEL VAN HET SPEL MEMORY

§ 2.1 Een beschrijving van het spel

Memory wordt (meestal door kinderen) gespeeld met een even aantal kaarten, waarvan de achterkanten niet van elkaar te onderscheiden zijn; de voorkanten zijn twee aan twee aan elkaar gelijk. Het aantal deelnemers is in principe willekeurig.

Voor het begin van het spel worden de kaarten geschud en met de achterkant naar boven op tafel gelegd. De speler die begint moet twee kaarten omdraaien. Als de kaarten gelijke voorkanten hebben mag hij ze houden, anders worden de kaarten op hun oorspronkelijke plaats teruggelegd (met de achterkant naar boven). Zolang hij twee gelijke kaarten pakt blijft de speler aan de beurt, de speler naast hem krijgt de beurt bij ongelijke kaarten. Het spel gaat net zo lang door, totdat geen kaarten meer op tafel liggen. Degene die dan de meeste paren heeft verzameld is de winnaar.

§ 2.2. Ons model van het spel

Zoals de naam het al zegt, legt het spel Memory de nadruk op het geheugen van de spelers. Het spel wordt meestal niet opgevat als een spel waarin het om taktisch inzicht gaat. Dat laatste zullen we wel doen.

Om het geheugen-aspect kwijt te raken zullen we veronderstellen dat de spelers een perfect geheugen hebben. Alle kaarten die eenmaal zijn omgedraaid blijven bekend. (In onze versie van het spel hoeft een eenmaal opengedraaide kaart dus niet terug-gedraaid te worden).

Verder veronderstellen we dat er maar twee spelers zijn.

Uiteraard is dit model één van de velen die mogelijk zijn. Voor variaties wordt verwezen naar Hoofdstuk 5.

Een taalkundige opmerking. In het vervolg zullen we met de term <u>nevenkaart</u> van een kaart de kaart bedoelen die een gelijke voorkant heeft als de kaart in kwestie. We excuseren ons voor het feit dat het woord "nevenkaart" niet in het woordenboek voorkomt.

HOOFDSTUK 3: ANALYSE VAN HET SPEL MET PERFECT GEHEUGEN

Dit hoofdstuk beginnen we met een algemene uitleg van het dynamisch programmeren (§3.1). Vervolgens komt de toepassing ervan op het spel Memory aan de orde (§3.2). Daarvoor worden de verschillende taktieken die er mogelijk zijn onderzocht (§3.3). Als we eenmaal de methode voor de computerberekening voldoende hebben ontwikkeld (§3.4) worden de resultaten gepresenteerd (§3.6). Het eind van dit hoofdstuk is gewijd aan de theoretische onderbouwing van de resultaten (§3.7). Er is ook een paragraaf gewijd aan de complexiteit van de berekeningen (§3.5).

§3.1 Het principe van het dynamisch programmeren.

Dynamisch programmeren biedt een eenvoudige oplossingsmethode voor problemen die gekenmerkt worden door een reeks van beslissingen. Gevraagd wordt die reeks van beslissingen zodanig te bepalen dat een zekere winstfunctie wordt gemaximaliseerd. In zo'n probleem worden een begin- en een eindtoestand onderscheiden en nul of meer tussentoestanden.

Een toestand wordt gekarakteriseerd door één of meer toestandsvariabelen, die we onder kunnen brengen in een toestandsvector $\underline{t}$. We noemen de begintoestand $\underline{t}_n$ en de eindtoestand $\underline{t}_0$. Bij elke toestand $\underline{t}_i$ $(1 \le i \le n)$ moet een beslissing b_i worden genomen. In het algemeen hebben we de keuze uit meer dan één beslissing in iedere toestand. De vector van de n achtereenvolgende beslissingen noemen we $\underline{b}^n$. De genoemde winstfunctie, w, is een functie van

~~verzameling van~~ de (begin)toestanden T en de te nemen beslissingen B.
Dus. $w: T \times B \to \mathbb{R}$
(i.p.v. $\mathbb{R}$ kunnen we ook met een andere verzameling te maken hebben).
Ons probleem is dus $\underline{b}^n$ zodanig te bepalen dat $w(\underline{t}_n, \underline{b}^n)$ maximaal is.

Stel dat we voor alle $\underline{t}_{n-1}$, d.w.z. alle toestanden die vanuit $\underline{t}_n$ na één beslissing worden bereikt, de reeks van beslissingen $\underline{b}^{n-1}$ kennen, die ervoor zorgen dat $w(\underline{t}_{n-1}, \underline{b}^{n-1})$ maximaal is. Dan geldt volgens het dynamisch programmeren (het principe van Bellman) voor de maximale $w(\underline{t}_n, \underline{b}^n)$:

$$w(\underline{t}_n, \underline{b}^n) = w'(\underline{t}_n, b_n) + w(\underline{t}_{n-1}, \underline{b}^{n-1}), \text{ met}$$

b_n zodanig dat het rechterlid maximaal is.
Hierin is w' een "lokale" winstfunctie, die aangeeft hoe groot de winst is die hoort bij de overgang tussen toestand $\underline{t}_n$ en toestand $\underline{t}_{n-1}$. (De overgang wordt bewerkstelligd door beslissing b_n).
Omdat we $w'(\underline{t}_n, b_n)$ en $w(\underline{t}_{n-1}, \underline{b}^{n-1})$ kennen zijn we in staat b_n zodanig te kiezen dat hun som maximaal wordt en hebben we ons probleem opgelost.

In het algemeen kennen we echter $w(\underline{t}_{n-1}, \underline{b}^{n-1})$ niet. Dit is geen bezwaar omdat we nu een soortgelijk probleem hebben als ons oorspronkelijk probleem. Dit probleem is echter minder gecompliceerd omdat we één beslissing minder te nemen hebben. Voor dit probleem kunnen we weer het principe van Bellman gebruiken. Herhaald toepassen van het principe leidt tenslotte tot:

$$w(\underline{t}_1, \underline{b}^1) = w'(\underline{t}_1, b_1) + w(\underline{t}_0, \ ), \text{ met } b_1$$

zodanig dat het rechterlid maximaal is.

$w(\underline{t}_0, \ )$ heeft geen tweede argument omdat in de eindtoestand geen beslissing meer behoeft te worden genomen. De winst voor iedere $\underline{t}_0$ moet gegeven zijn. (Merk verder op dat hier geldt $\underline{b}^1 = b_1$ omdat we met $\underline{b}^1$ een vector met 1 component aangeven).

Voor iedere toestand $\underline{t}_1$ kan de bijbehorende optimale beslissing $\underline{b}^1$ worden bepaald zodat $w(\underline{t}_1, \underline{b}^1)$ maximaal is. Door het pad terug te volgen wordt tenslotte de optimale beslissingenreeks $\underline{b}^n$ bepaald zodat $w(\underline{t}_n, \underline{b}^n)$ maximaal is.

§3.2. Toepassing van dynamisch programmeren op het spel Memory

§3.2.1 De toestandvector

In het spel met perfect geheugen blijken er drie noodzakelijke en voldoende toestandvariabelen te zijn:

1. Het aantal nog in het spel zijnde kaarten: K.
   (K is altijd even)
2. Het aantal reeds bekende kaarten B.
   (d.w.z. het aantal kaarten dat minstens één keer is omgedraaid)
3. Het aantal gepaarde bekende kaarten. P.
   (bij P worden alle bekende kaarten geteld waarvan ~~ook~~ de nevenkaart bekend is. P is dus altijd even.)

In het vervolg zullen we een toestand als een drietal $\langle K, B, P \rangle$ noteren.

Dit is niet de enige toestandsvoorstelling. Er zijn er vele die ermee equivalent zijn: men kan b.v. het aantal onbekende kaarten tellen i.p.v. de bekende of men kan de grootheden k en P vervangen door hun helft, omdat ze toch altijd even zijn.

Er is nog één onafhankelijke grootheid die als toestands-variabele in aanmerking komt, namelijk wie van de twee spelers er aan de beurt is. Strikt genomen kan deze grootheid niet gemist worden. Ze is echter van een hele andere aard dan de grootheden k, B en P. In de hierna volgende analyse beschouwen het probleem steeds voor de speler die aan de beurt is. Beurtverlies wordt voorgesteld door complementeren van de winst t.o.v. k. (De som van de winst van beide spelers is altijd k.) De beurt is dus niet expliciet maar wel impliciet in de berekeningen terug te vinden.

Het is duidelijk dat voor k, B en P moet gelden:

$$k \geq B \geq P \geq 0.$$

Dit blijkt echter niet de enige beperking te zijn die aan de toestandsvariabelen wordt opgelegd. De bovengrens van B is niet k maar $\frac{k+P}{2}$. Dit is als volgt in te zien:

B = "het aantal gepaarde bekende kaarten" + "het aantal ongepaarde bekende kaarten".

$$B_{max} = P + \frac{k-P}{2} = \frac{k+P}{2}$$

(B_{max} geldt als van alle ongepaarde kaarten $(k-P)$ de helft bekend is).

We krijgen dus als randvoorwaarde voor de toestand:

$$k \geq \frac{k+P}{2} \geq B \geq P \geq 0 \qquad (3.1)$$

§ 3.2.2 De winstfunctie

In tegenstelling tot hetgeen in § 3.1 is gezegd, ligt bij het spel Memory de bij een beslissing behorende winst niet eenduidig vast. Bij het aandraaien van een onbekende kaart (dit kan onderdeel zijn van een beslissing) kan men in het algemeen in meerdere toestanden terecht komen afhankelijk van de afbeelding op de voorkant. Er kan bijvoorbeeld een nieuw paar bekende kaarten ontstaan, of niet. We moeten daarom een toestandsovergang met de erbij behorende kans wegen en spreken daarom niet van de winst, maar van de verwachte winst.

In het vervolg geven we met $w(k,B,P)$ de maximale verwachte winst in toestand $\langle k,B,P \rangle$ aan. De beslissingsvector komt dus niet in het argument voor omdat we alleen de maximale winst beschouwen.
Met $w_i(k,B,P)$ geven we de winst aan die beslissing b_i in toestand $\langle k,B,P \rangle$ oplevert. In het algemeen geldt voor w_i:

$$w_i(k,B,P) = \sum_{i=1}^{m} p_i \left[w(k-2,B_i,P_i) + 2 \right] + \sum_{j=1}^{n} p_j \left[k - w(k,B_j,P_j) \right]$$

In de formule is aangegeven dat we de toestandsovergangen in twee groepen kunnen verdelen.

1. De speler aan de beurt pakt twee kaarten en blijft aan de beurt. Het aantal kaarten in het spel is nu k-2.
2. De speler aan de beurt verliest zijn beurt.

De p_i en p_j stellen de kansen voor op de overgangen naar de toestanden $\langle k-2,B_i,P_i \rangle$ respectievelijk $\langle k,B_j,P_j \rangle$.
Uiteraard moet gelden: $\sum_{i=1}^{m} p_i + \sum_{j=1}^{n} p_j = 1$.

VERSLAG / RAPPORT PROEFNR. NAAM:

VERVOLGVEL NR. 10 DATUM:

Tenslotte moet gelden dat alle toestanden waarin men over kan gaan minder complex zijn dan de oorspronkelijke toestand. Dit is essentieel voor dynamisch programmeren. Het zal later blijken dat dit voor alle beslissingen (op één na, waarin de toestand onveranderd blijft) het geval is.

De maximale winst voor de toestand <K,B,P> wordt als volgt bepaald:

$$w(K,B,P) = \max \{ w_1(K,B,P), \ldots, w_\ell(K,B,P) \}$$

(Er staat hier niets anders dan het principe van Bellman)
Het getal ℓ slaat op het aantal mogelijke beslissingen in toestand (K,B,P). Bij het spel Memory, waarin we met eindige en discrete grootheden te maken hebben, is ℓ altijd eindig.

In de volgende paragraaf gaan we uitvoerig in op de mogelijke beslissingen die we voortaan taktieken zullen noemen.

§ 3.3 De mogelijke taktieken.

§ 3.3.1 Een uitputtende analyse van alle taktieken

In deze paragraaf zullen alle mogelijke taktieken (dus niet alleen die taktieken die zinvol lijken) worden beschouwd.

Voor de eerste kaart die wordt omgedraaid hebben we, in het algemeen, de keuze uit drie soorten kaarten (er worden hier enkele afkortingen ingevoerd die notaties verderop maken vereenvoudigen):

1. bmn = een bekende kaart met bekende nevenkaart

VERSLAG / RAPPORT PROEFNR. NAAM:

VERVOLGVEL NR. 11 DATUM:

2. bzn = een bekende kaart zonder bekende nevenkaart
3. onb = een onbekende kaart

Voor de tweede kaart hebben we alle mogelijkheden die hierboven zijn genoemd plus nog een vierde mogelijkheid

4. nek = de nevenkaart van de eerste kaart

(Als de eerste kaart een bmn is dan betekent bmn voor de tweede kaart dus niet dat er een paar wordt opgepakt, maar dat van twee verschillende bekende paren een kaart wordt gepakt)

We spreken af dat de verzameling K_1 de mogelijkheden voor de eerste kaart bevat en dat de verzameling K_2 de mogelijkheden voor de tweede kaart. Dus
$K_1 = \{bmn, bzn, onb\}$
$K_2 = \{bmn, bzn, onb, nek\}$

Als we voor de eerste kaart bmn of bzn kiezen dan is het resultaat van het omdraaien gedetermineerd. De keuze voor de tweede kaart kan niet afhangen van het resultaat van de eerste. Als we echter met onb beginnen, dan is het resultaat ervan wel degelijk van invloed op de keuze van de tweede kaart. We voeren daarom het predikaat "kbn" (kaart met bekende nevenkaart) in met als eigenschap:

kbn = "waar" als de nevenkaart van de eerste kaart al bekend is,
kbn = "onwaar" als de nevenkaart van de eerste kaart niet tot de bekende kaarten behoort.

Merk op, dat we met dit predikaat alle mogelijke uitkomsten

van onb voor de eerste kaart kunnen beschrijven.
Het predikaat wordt gebruikt in de volgende constructie:

ALS kbn DAN p ANDERS q met $p \in K_2$ en $q \in K_2 | \{nek\} (=K_1)$
(als kbn="onwaar" dan is nek niet gedefinieerd)
en ook $p \neq q$ (anders is er geen sprake van een van de eerste kaart afhankelijke keuze).

Alle mogelijke voorkomens van de bovenstaande constructie brengen we onder in de verzameling L_2. Dus:

$L_2 = \{$ ALS kbn DAN p ANDERS q $| \; p \in K_2 \wedge q \in K_1 \wedge p \neq q \}$

Tenslotte definieren we een taktiek als een tweetal $\langle a, b \rangle$. Er moet gelden $a \in K_1$ en $b \in K_2 \cup L_2$, waarbij nog aan de volgende randvoorwaarden moet worden voldaan:

1. Als a=bmn dan is $b \in K_2$
2. Als a=bzn dan is $b \in K_2 | \{nek\} (=K_1)$
3. Als a=onb dan is $b \in (K_2 \cup L_2) | \{nek\} \; (=K_1 \cup L_2)$

Randvoorwaarde 1 geeft aanleiding tot $|K_2| = 4$ taktieken.
" 2 " " $|K_1| = 3$ "
" 3 " " $|K_1| + |L_2| = 12$ "
(Voor L_2 geldt immers $|L_2| = |K_1| \times |K_2| - |K_1| = 9$)
(Met $|X|$ wordt bedoeld het aantal elementen van verzameling X).
Er zijn dus in totaal 19 verschillende taktieken te onderscheiden.

In de volgende paragrafen kijken we naar de verwachte winst die elk van deze taktieken opleveren. Het zal blijken dat sommige gelijke winsten opleveren, zodat ze samengenomen kunnen worden.

§ 3.3.2. De berekening van de verwachte winst voor twee taktieken

In deze paragraaf zal voor twee taktieken de verwachte winst worden berekend, om de methode van aanpak duidelijk te maken. In de volgende paragraaf zal dan zonder toelichting een opsomming worden gegeven van alle taktieken die tot verschillende winsten leiden.

voorbeeld 1 $\langle$bmn, nek$\rangle$ (pak een paar bekende kaarten op)
Randvoorwaarde: $P \geq 2$ (er moet minstens één paar bekende kaarten zijn).

$$w_1(K,B,P) = w(K-2, B-2, P-2) + 2.$$

voorbeeld 2 $\langle$onb, ALS kbn DAN nek ANDERS bzn$\rangle$
(draai een onbekende kaart en als de nevenkaart ervan bekend is, pak die dan; draai anders een andere bekende kaart)
Randvoorwaarde: $K \geq B+1$ (er is minstens één onbekende kaart)
en $B \geq P+1$ (de kans dat zijn nevenkaart bekend is, is groter dan nul)
en $B \geq 2$ (er moeten minstens twee bekende kaarten zijn, anders is deze taktiek equivalent aan $\langle$onb, bzn$\rangle$)

$$w_2(K,B,P) = \frac{B-P}{K-B}\left[w(K-2, B-1, P) + 2\right] + \frac{K+P-2B}{K-B}\left[K - w(K, B+1, P)\right]$$

Hierin stelt $\frac{B-P}{K-B}$ de kans voor, dat de nevenkaart van de eerste kaart bekend is. (Het aantal ongepaarde bekende kaarten gedeeld door het aantal onbekende kaarten; het aantal ongepaarde bekende kaarten is even groot als het aantal ongepaarde onbekende kaarten).
In dat geval kunnen twee kaarten worden opgepakt

en komt men in een toestand terecht, waarin het aantal kaarten met twee is verminderd en het aantal bekende kaarten met één.
$\frac{K+P-2B}{K-B} = 1 - \frac{B-P}{K-B}$ is het complement van de eerste kans. Omdat er geen twee kaarten opgepakt kunnen worden verliest men de beurt en komt men in een toestand terecht waarin het aantal bekende kaarten met één is toegenomen.

§ 3.3.3 Opsomming van alle taktieken met verschillende verwachte winsten

Voordat de taktieken en hun verwachte winst worden opgesomd, worden eerst alle in de formules te gebruiken kansfactoren op een rijtje gezet en van een naam (P_i) voorzien. Dat zal de leesbaarheid van de formules vergemakkelijken.

P_1 t/m P_4 hebben betrekking op de eerste keer dat er getrokken wordt uit de onbekende kaarten. Ze hebben als noemer $K-B$: het aantal onbekende kaarten. P_5 t/m P_9 hebben betrekking op de tweede keer, het aantal onbekende kaarten is nu met één verminderd.

Naam	Uitdrukking	Toelichting
P_1	$\frac{B-P}{K-B}$	De kans op een kaart met bekende nevenkaart
P_2	$\frac{K+P-2B}{K-B}$	$1-P_1$.
P_3	$\frac{1}{K-B}$	De kans op één bepaalde onbekende kaart
P_4	$\frac{B-P-1}{K-B}$	$P_1 - P_3$.
P_5	$1/K-B-1$	zie P_3, nu voor de tweede trekking
P_6	$(B-P)/K-B-1$	de kans op een kaart waarvan de nevenkaart bekend is behalve die ene uit P_5.

Naam	Uitdrukking	Toelichting
P_7	$\frac{K+P-2B-2}{K-B-1}$	$1-P_5-P_6$
P_8	$(B-P-1)/K-B-1$	De kans dat de tweede kaart een bekende nevenkaart heeft als de eerste ook een bekende nevenkaart he
P_9	$(K+P-2B)/K-B-1$	$1-P_8$

Van de taktieken die zijn samengevoegd geven we de bestanddelen die van verschillende taktieken afkomstig zijn, boven elkaar en tussen accolades weer. Bijvoorbeeld: $\left\langle \begin{Bmatrix} bmn \\ bzn \end{Bmatrix}, \begin{Bmatrix} bmn \\ bzn \end{Bmatrix} \right\rangle$ is de samenvoeging van de vier taktieken $\langle bmn, bmn\rangle$, $\langle bmn, bzn\rangle$, $\langle bzn, bmn\rangle$ en $\langle bzn, bzn\rangle$.
In het geval van samenvoeging is de randvoorwaarde de "of" van de randvoorwaarden van de samenstellende taktieken.
Ook de taktieken zijn van een naam (T_i) voorzien. Hier volgen ze:

$T_1 = \langle bmn, nek\rangle$
Randvoorwaarde: $P \geq 2$
$w_1(K,B,P) = w(K-2, B-2, P-2) + 2.$

$T_2 = \left\langle \begin{Bmatrix} bmn \\ bzn \end{Bmatrix}, \begin{Bmatrix} bmn \\ bzn \end{Bmatrix} \right\rangle$ = "pas".
Randvoorwaarde: $B \geq 3$ of $B \geq 2 \wedge P = 0$.
$w_2(K,B,P) = K - w(K,B,P) = K - w_2(K,B,P) = \frac{K}{2}$
(zie ook § 3.4).

$T_3 = \langle bmn, onb\rangle$ of $\langle onb, bmn\rangle$ of
$\left\langle onb, \text{ALS } kbn \begin{Bmatrix} \text{DAN } bmn \text{ ANDERS } bzn \\ \text{DAN } bzn \text{ ANDERS } bmn \end{Bmatrix} \right\rangle$