

Qubic opnieuw opgelost

Schoo, P.N.A.; van den Herik, H.J.

Published in:

Convex Courier: Samenwerkingsverband van de universitaire rekencentra Rotterdam, Tilburg, Maastricht

Publication date:

1992

[Link to publication](#)

Citation for published version (APA):

Schoo, P. N. A., & van den Herik, H. J. (1992). Qubic opnieuw opgelost. *Convex Courier: Samenwerkingsverband van de universitaire rekencentra Rotterdam, Tilburg, Maastricht*, 8(6), 13-16.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright, please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Patrick Schoo en
Jaap van den Herik
(vakgroep informatica, RL)

QUBIC

opnieuw

opgelost

Qubic is een spel dat wordt gespeeld door twee personen op een drie-dimensionaal bord. Het spel bestaat uit een kubus van 4x4x4 velden

plus 64 stenen waarvan er 32 wit en 32 zwart zijn. Beide spelers plaatsen om de beurt een steen van hun eigen kleur op één van de lege velden. Degene die er als eerste in slaagt vier stenen op één rij (horizontaal, vertikaal of diagonaal) te plaatsen wint het spel. Het spel qubic is inmiddels van de markt genomen. Tot voor kort werd het spel echter nog gespeeld in competitieverband, en wel op de 2e en 3e Computer Olympiade in 1990 en 1991. Voor de laatste Olympiade (Maastricht, 1991) hebben Victor Allis en Patrick Schoo het programma Qbig geschreven dat met wit altijd wint (d.w.z. de speltheoretische waarde bereikt) en met zwart zeer sterk tegenspel biedt. Dit is de reden dat qubic in 1992 niet meer terug zal keren op de 4e Computer Olympiade, die gehouden wordt in Londen van 5-11 augustus.

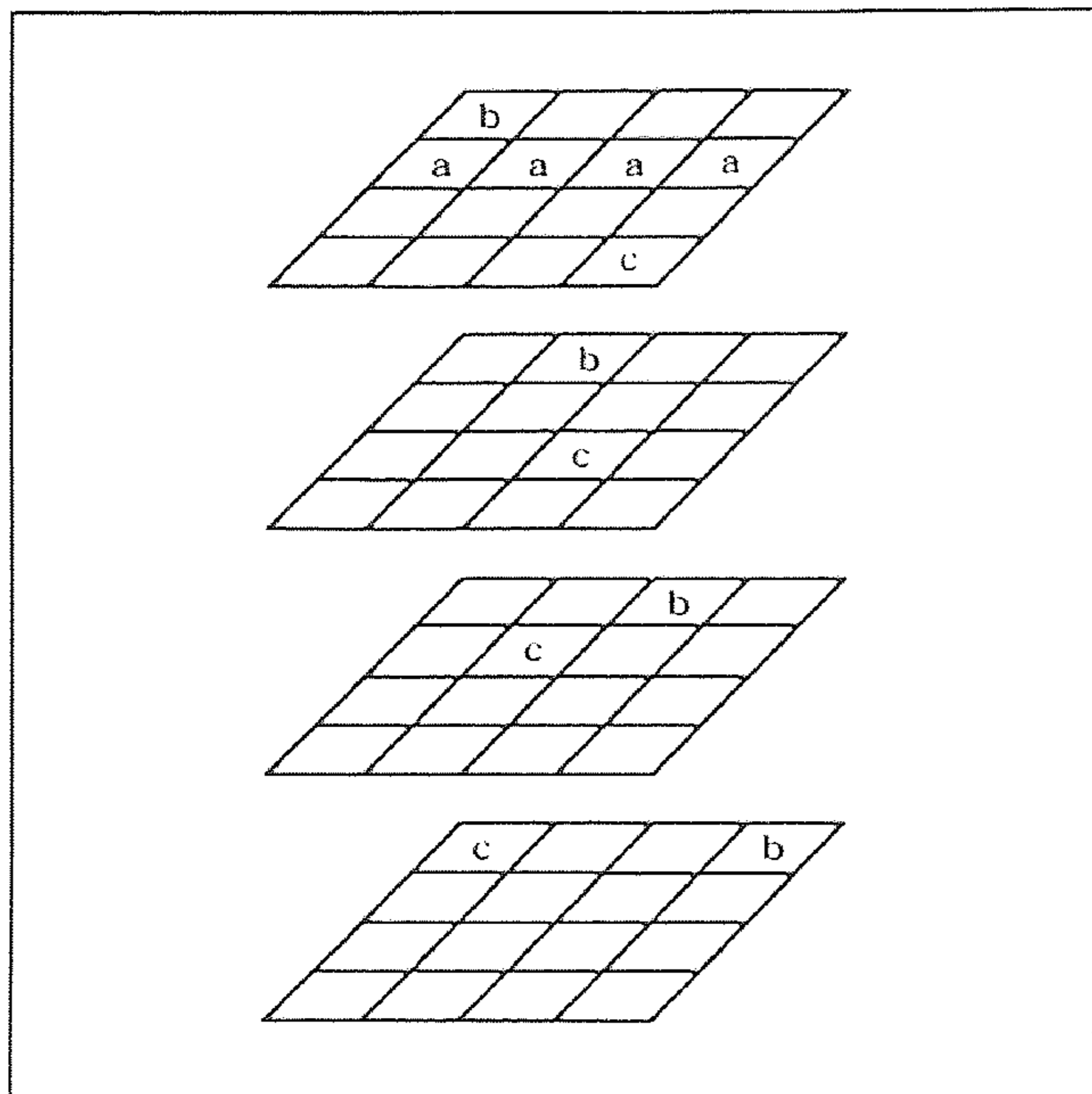
De historie van qubic gaat vele jaren terug. In 1980 heeft Oren Patashnik aangetoond dat qubic altijd gewonnen is voor de witspeler tegen elke verdediging van zwart. Helaas werd dit resultaat in een tamelijk onbekend tijdschrift gepubliceerd, zodat het aan de aandacht ontsnapte van veel onderzoekers die op het terrein van computerspelen werkzaam zijn. Toch vormde het werk van Patashnik, ook toen het eenmaal bekend was, geen aanleiding om qubic te verwijderen van de Computer Olympiade. Het is namelijk niet voldoende om aan te tonen wat de speltheoretische waarde van een bepaald spel is. Men moet ook over een speelprogramma beschikken waarmee wit altijd wint (en dat liefst nog redelijk speelt met zwart). Patashnik had het spel slechts opgelost en geen speelprogramma geschreven.

In dit artikel geven we een beschrijving van de methodieken die gebruikt zijn om het spel qubic opnieuw op te lossen, en ook om een spelend programma te schrijven. Natuurlijk is een grote hoeveelheid reken capaciteit nodig om alle varianten door te rekenen, maar alleen rekenkracht is niet voldoende. Twee zoekstrategieën spelen een belangrijke rol in het reduceren van het aantal varianten dat moet worden doorzocht: threat-space search en proof-number search. De eerste methode wordt hieronder behandeld; voor de tweede verwijzen we naar Allis et al. (1991).

Analyse van Qubic

Voor het schrijven van een goed speelprogramma is het belangrijk het spel grondig te analyseren. Na de analyse wordt bepaald welke bestaande technieken bruikbaar zijn en, indien nodig, welke nieuwe technieken ontwikkeld dienen te worden. Bij qubic beginnen we de analyse aan het einde van het spel. Dat is namelijk als één van

de spelers vier stenen op een rechte lijn kan plaatsen. Zo'n rechte lijn noemen we een *groep*. In totaal zijn er 76 verschillende groepen op een leeg bord. In figuur 1 laten we drie verschillende soorten groepen zien. Groep a (een horizontale groep in het bovenvlak) is een ééndimensionale groep, waarvan er $16 \times 3 = 48$ zijn. We kijken in



Figuur 1: drie verschillende typen groepen

drie richtingen: lengte, breedte en hoogte. Op elk vlak kunnen we 4 groepen van links naar rechts vormen, wat samen 16 groepen maakt. Evenzo hebben we 16 groepen die van voor naar achteren lopen, en 16 die van boven naar beneden lopen. Groep b (een diagonale groep in het achtervlak) is een tweedimensionale groep, waarvan er $8 \times 3 = 24$ zijn. Op elk vlak kunnen we 2 diagonale groepen vormen, voor een totaal van 8. Bovendien is de kubus op drie manieren op te delen in vlakken. Groep c (een lichaamsdiagonaal) is een driedimensionale groep, waarvan er 4 aanwezig zijn; de lichaamsdiagonalen van de kubus.

Als een speler drie velden van een groep heeft opgevuld, en het vierde is nog leeg, dan dreigt hij te winnen. Als de tegenstander niet zelf kan winnen dan is hij verplicht om op het vierde veld te spelen. Immers doet hij dat niet dan verliest hij op de volgende zet. Het opvullen van drie velden in een groep is dus een *directe* dreiging voor de tegenstander. Het spel komt meestal tot een einde als één van de spelers een serie van dreigingen uitvoert, waarbij uiteindelijk een situatie wordt gecreëerd waarin de tegenstander twee dreigingen tegelijk moet pareren. In figuur

2 geven we een voorbeeld van zo'n serie in een vlak. Wit heeft al drie stenen in dit vlak geplaatst (namelijk in de drie hoekpunten, zie figuur 3), terwijl zwart op de andere vlakken heeft gespeeld. Wit speelt nu steen 1, en zwart moet antwoorden met 2. Dit proces van dreigen en pareren herhaalt zich op de zetten 3 tot 8, totdat zet 9 wordt uitgevoerd. Nu moet zwart tegelijk op a en b antwoorden wat natuurlijk niet gaat, zodat wit hier wint.

Threat-space search

Threat-space search is toegespitst op het vinden van een serie dreigingen die tot winst leidt. Het is mogelijk om een eenvoudige zoekmethode te gebruiken om de winst te vinden, maar deze zal veel varianten doorzoeken die niet ter zake doen.

Dit betekent dat die methode eveneens langzaam zal zijn, en dat willen we juist vermijden. De werking van het threat-space search algoritme illustreren we aan de hand van figuur 3. We zien dat in deze stelling zes dreigzetten uitgevoerd kunnen worden (b3, c2, b4, c4, d2, d3). Stel dat we zet b3 uitvoeren, dan moet de tegenstander antwoorden op c2. Misschien kan de steen op b3 later gebruikt worden om nieuwe dreigingen te creëren?

De prijs voor deze zet is evenwel dat zwart veld c2 in handen heeft gekregen. Elke dreigzet levert ons dus iets op, maar het kost ons ook wat. Evenzo kunnen we veld c2 in handen krijgen, maar dat gaat weer ten koste van veld b3. Dreigzetten komen altijd in paren voor. Alle

○	1	2	○
a	3	7	9
	5	4	b
8	6		○

Fig. 2: winst in 11 zetten

○			○	4
				3
				2
			○	1
a	b	c	d	

Figuur 3: beginstand met winst in 11 zetten

directe dreigingen zijn weergegeven in tabel 1. De tweede kolom geeft aan welk veld wordt bezet, de derde kolom welk veld de tegenstander krijgt, en de laatste kolom de twee andere velden die nodig zijn om de dreiging te vormen. In figuur 4 hebben we de directe dreigzetten gemarkeerd met een 1. De nullen geven de stenen aan die reeds gespeeld zijn.

Nummer	Veld	Kost	Benodigd
1	b3	c2	a4, d1
2	c2	b3	a4, d1
3	b4	c4	a4, d4
4	c4	b4	a4, d4
5	d2	d3	d1, d4
6	d3	d2	d1, d4

Tabel 1: Directe dreigingen

Tabel 1 en figuur 4 gebruiken we vervolgens om te onderzoeken of we de velden die we verkregen hebben na één dreiging met elkaar kunnen combineren tot een nieuwe dreiging. Bezien we bijvoorbeeld de velden b3 en b4 dan valt op dat ze mogelijk te gebruiken zijn voor dreigingen via b1 of b2. Deze nieuwe (indirecte) dreigingen vormen de tweede laag in de zoekruimte. In tabel 2 hebben we alle *indirecte* dreigingen op dezelfde manier weergegeven als in tabel 1 de *directe* dreigingen.

Nummer	Veld	Kost	Benodigd
7	b1	b2	b3, b4
8	b2	b1	b3, b4
9	c1	c3	c2, c4
10	c3	c1	c2, c4
11	a2	b2	c2, d2
12	b2	a2	c2, d2
13	a3	c3	b3, d3
14	c3	a3	b3, d3

Tabel 2: De tweede laag

Tevens hebben we de nieuwe dreigingen in figuur 5 aangegeven met het getal 2. Hier zien we iets interessants: de velden b2 en c3 bevatten tweemaal het getal 2. Dit betekent dat deze velden op twee manieren via dreigingen geclaimd kunnen worden. Als we even in herinnering roe-

pen dat ons doel niet is om zoveel mogelijk dreigingen te creëren, maar om een dubbele dreiging te creëren, dan is duidelijk dat zo'n veld met twee getallen een potentiële dubbele dreiging is. Om te zien of dat ook werkelijk zo is moeten we de twee dreigingen aan een nader onderzoek onderwerpen. We beginnen met veld b2. In de kolom 'veld' van tabel 2 komt deze tweemaal voor. Bij de eerste afleiding voor b2 (regel 8), geven we b1 weg aan de tegenstander. Voor b2 hebben we tevens b3 en b4 nodig. Die zoeken we op in de eerste tabel. De kosten voor deze twee zetten zijn resp. c2 (regel 1) en c4 (regel 3).

In figuur 6 hebben we de gespeelde velden en de velden die we nodig hebben voor de dubbele dreiging op b2 aangegeven met een '+'; op de velden die we aan de tegenstander geven noteren we een '-'. Verder is in figuur 6 zowel de eerste (regel 8) als de tweede afleiding voor b2 (regel 12) opgenomen. Nu zien we plotse-ling een conflict op de velden b3 en c2. Voor regel 12 en de daarbij behorende regel 2 moeten we veld b3 aan de tegenstander geven, terwijl we voor de regel 8 veld b3 juist zelf nodig hebben. Het is duidelijk dat we nog geen winst gevonden hebben. Na veld b2 moeten we ook veld c3 (zie figuur 5) natrekken. Zonder uitleg vermelden we dat ook bij het nagaan van c3 een conflict ontstaat. De eerste en tweede laag zijn blijkbaar niet voldoende om een winstpad te vinden. Daarom vervolgen we vanuit figuur 5 met startpunt a1 ons zoekproces met het genereren van een derde laag (tabel 3 en figuur 7). Ook hier worden de regels in pa-

4	0	1	1	0
3		1		1
2			1	1
1				0
	a	b	c	d

Figuur 4

4	0	1	1	0
3	2	1	$\frac{2}{2}$	1
2	2	$\frac{2}{2}$	1	1
1		2	2	0
	a	b	c	d

Figuur 5

4	+	+	-	+
3		+		-
2	-		+	+
1		-		+
	a	b	c	d

Figuur 6

4	0	1	1	0
3	2	1	$\frac{2}{2}$ 3	1
2	2	$\frac{2}{2}$	1	1
1	$\frac{3}{3}$	2	$\frac{2}{3}$	0
	a	b	c	d

Figuur 7

Nummer	Veld	Kost	Benodigd
15	a1	c1	b1, d1
16	c1	a1	b1, d1
17	a1	c3	b2, d4
18	c3	a1	b2, d4

Tabel 3: De derde laag

Na drie regels hebben we het volgende resultaat. De beide afleidingen voor a1 (regel 15, 17) en c1 (regel 9, 16) leiden weer tot een conflict. Echter, de nieuwste afleiding voor c3 (regel 18) gecombineerd met afleiding 14 leidt tot succes getuige figuur 8. In deze figuur is geen enkel veld bezet met zowel een '+' als een '-'. De enige manier waarop ons bewijs nog stuk kan lopen is als de tegenstander tijdens het verdedigen toevallig een eigen dreiging creëert. Dat is de laatste test die nog uitgevoerd moet worden. In ons voorbeeld is eenvoudig aan figuur 8 te zien dat er van vijandelijke dreigingen geen sprake is. Nergens heeft de tegenstander meer dan twee stenen (aangegeven met '-') in een groep. Rest ons nog om uit het bewijs dat de positie gewonnen is een zettenreeks te onttrekken. De eenvoudigste manier is om de afleidingsregels die van toepassing zijn in opsommende volgorde toe te passen. De volgorde wordt dan 1, 3, 6, 8, 14, 18. De zettenreeks is weergegeven in figuur 9.

4	+	+	-	+
3	-	+	+	+
2		+	-	-
1	-	-		+
	a	b	c	d

Figuur 8

	○	③	④	○
a		①	⑨	⑤
		⑦	②	⑥
b		⑧		○

Figuur 9: De zettenreeks 1, 3, 6, 8, 14, 18

ren gegenereerd. Bij elke regel wordt onmiddellijk nagegaan of deze tot een oplossing leidt.

pliceerd zijn, maar waarin geen winstmogelijkheden zitten. Threat-space search heeft snel 'beken' dat winst niet mogelijk is terwijl een conventionele zoekmethode daar veel langer over doet.

In het volgende nummer van de Convex Courier beschrijven we hoe het threat-space algoritme als basis is gebruikt voor het oplossen van qubic. Tevens geven we een korte impressie van het qubic toernooi op de 3e Computer Olympiade.

Referenties

Allis, L.V., van der Meulen, M. en van den Herik, H.J. (1991). *Proof-Number Search*. Report CS 91-01. University of Limburg, Maastricht, The Netherlands.

Patashnik, O. (1980). Qubic: 4x4x4 Tic-Tac-Toe. *Mathematics Magazine*, Vol. 53, pp. 202-216.

Er zijn dus in totaal 18 afleidingen nodig om een winst van 11 zetten te vinden. Andere zoekmethoden vinden deze winst ook wel na het doorzoeken van een paar honderd stellingen, maar volgens ons blijkt de kracht van threat-space search juist uit stellingen die weliswaar gecom-