Enige wiskundige hulpmiddelen

De specificatie van een systeem, programma of enig ander artefact beschrijft precies **wat** het moet doen of zijn. Het is daarbij het formele referentiepunt voor systeemconstructie. Verificatie (bewijs dat het artefact aan de specificatie voldoet) moet mogelijk zijn en dus worden hoge mathematische eisen gesteld aan de specificatie, haar verfijningen en de vervullingen daarvan. Het belang van eenduidige en bewijsbare specificaties is zelfs zo groot, dat er vele formele specificatietalen en -methoden bestaan. Binnen projecten zijn de specificateurs vaak gehouden zich van een dergelijke afgesproken taal en/of methode te bedienen. Niet zelden is dat een taal die vele geautomatiseerde ontwerp- en verificatiemogelijkheden heeft, zodat de menselijke verificatiefouten vervangen worden door machinale.

In het onderhavige specificatieproject staan we in feite alle wiskunde toe, dus ook en met name de wiskunde zoals die op de middelbare school is geleerd. Echter, we hebben daaraan wellicht niet helemaal genoeg. We bespreken daarom in hoog tempo enige additionele mathematische structuren die van nut kunnen zijn voor de mathematische specificaties. Aan de orde zullen komen verzamelingen, rijtjes en functies terwijl we terloops enige notatie zullen introduceren. De notatie eisen zijn echter niet terloops! Mathematische expressies moeten met mathematische precisie genoteerd worden, beschouw het maar als een taal waarin taalfouten desastreus zijn.
Dit stukje is vooral bedoeld als technische ondersteuning om de OGO-opdracht te kunnen vervullen. Voor de echte kennis dezer dingen verwijzen we onder andere naar het vak Logica en verzamelingenleer.

1 Bulktypen

Verzamelingen en rijtjes zijn zogenaamde *bulktypen*, waarin van een gegeven basistype meerdere exemplaren bijeengegaard zijn. Verzamelingen zijn bulks met van elk ding ten hoogste één exemplaar, rijtjes zijn bulks die per ding meerdere exemplaren mogen hebben. Bij rijtjes is de volgorde waarin we die dingen bekijken van groot belang, bij verzamelingen niet.
Vaak wordt de postzegelverzameling als voorbeeld genomen. Als die in een schoenendoos zit is het geen rijtje, want de postzegels zitten er ongeordend in, maar ook is het vaak geen verzameling omdat er per postzegel meerdere exemplaren in mogen zitten (ruilmateriaal). Als ze in een insteekalbum zitten gaat het om een rijtje, als het "enkele" zijn is het ook een verzameling. Als ze in een verkoopzakje zitten in de postzegelwinkel is het vaak een verzameling (serie zomerzegels 1952 zonder plakker of 60 stuks Monte Negro met opdruk), maar geen rijtje.
Andere voorbeelden van bulktypen zijn bomen en bags, maar die zullen we hier niet behandelen.

1.1 Rijtjes

We beschouwen een eindig rijtje over een "alfabet" A als een collectie dingen die er uitziet als $[a_0, a_1, \ldots, a_n]$ waarbij alle geënumereerde inwoners a_i uit A komen. Het *lege rijtje* wordt genoteerd als $[\,]$, en $[a]$ is het singleton rijtje met a als enige inwoner.
Twee rijtjes zijn gelijk als ze even lang zijn en op de zelfde posities dezelfde waarden hebben staan. De collectie van alle eindige rijtjes over A geven we wel aan met $\mathbb{L}(A)$, met de $\mathbb{L}$ van list, we spreken ook wel over lijsten in plaats van rijtjes.
Het voorkomen van een element x in een rijtje ar wordt genoteerd als $x \Leftarrow ar$.
Twee rijtjes ar en br kunnen via *catenatie* aan elkaar geplakt worden om zodoende weer een nieuw rijtje te vormen: $ar +\!\!+ br$. Behalve catenatie kunnen we ook *comprehensie* gebruiken om nieuwe rijtjes te construeren. Dat wil zeggen dat we andere rijtjes en selectie en functietoepassing op de bewoners van die rijtjes gebruiken om de nieuwe rij-bewoners te beschrijven:

$$dkr(ar) \;=\; [a \;:\; a \Leftarrow ar \;\text{ en }\; a \text{ is een kwadraat} \;:\; 2 * \sqrt{a}] \tag{1}$$

De functie *dkr* maakt van een rijtje ar een nieuw rijtje waarvan de beschrijving in de rechterkant staat. De expressie rechts bestaat uit drie delen:

declaratie Hier worden de hulpvariabelen gegeven die een rol spelen in het definiëren van de rij-elementen. Ze bestaan alleen binnen de expressie, daarbuiten dus niet. We noemen die hulpvariabelen daarom wel *dummies*. Het type van de dummies kan in het declaratiedeel erbij genoemd worden.

domein Dit legt het waardegebied van de dummies vast en in principe ook de wijze waarop het waardegebied doorlopen wordt. Als we niets eisen is elke waarde toegestaan. In dat geval schrijven we niets of als niets te weinig lijkt schrijven we `true` om aan te geven dat alles goed is.

term De rij-elementen staan hier als waarden, uitgedrukt in termen van de waarden van de dummies.

De volgorde van de nieuwe rij-elementen is dezelfde als die van de rij in het domein (voor zover die elementen worden gebruikt).
Dus de a in (1) is een dummy die alleen binnen de expressie hierboven bestaat en gebruikt wordt om de constructie van het nieuwe rij-element $2 * \sqrt{a}$ in de term te faciliteren. Die dummy staat voor elk getal dat in de rij ar staat, zolang dat getal maar een kwadraat is. Met dat getal maken we een nieuw rij-bewoner door de wortel daaruit te trekken en het resultaat met twee te vermenigvuldigen. Dus

$$dkr([0, 481, 169, -1, 37, 0, 1369]) = [0, 26, 0, 74] \;\text{ en }\; dkr([\,]) = [\,]$$

Bij rijtjes kunnen we ook nog gebruik maken van de volgorde waarin de bewoners in het rijtje zitten. Beschouw daartoe het rijtje ar als een functie die gedefinieerd is op een

beginstuk van $\mathbb{N}$, de natuurlijke getallen $\{0, 1, 2, 3, \ldots\}$. Dwz dat $ar(n)$ de rijtjesbewoner is met index n, het n+1-ste element van de rij ar. Als l de lengte is van ar (notatie: $l = \#ar$) kunnen we dus zeggen dat

$$ar \;=\; [i \;:\; 0 \leq i < l \;:\; ar(i)]$$

terwijl

$$\mathsf{rev}(ar) \;=\; [i \;:\; 0 \leq i < l \;:\; ar(l-1-i)] \tag{2}$$

natuurlijk weer een geheel ander rijtje is (met dezelfde bewoners maar in een andere volgorde). De volgorde in ar en $\mathsf{rev}(ar)$ wordt bepaald door de volgorde van de waarden van de dummy in het domein, die begint bij 0 en stapsgewijs groeit tot en zonder l.
Merk op dat $\mathsf{rev}(ar)$ op eenduidige wijze uit ar ontstaat, i.e. rev is een functie op rijtjes.

Opgave

1. Geef in goed Nederlands weer wat de volgende rijtjesexpressies voorstellen:
   (a) $[a : a \lessdot ar : 1]$
   (b) $[a : a \lessdot ar \;\wedge\; a < 0 : a] +\!\!+ [a : a \lessdot ar \;\wedge\; a > 0 : a]$
   (c) $[i \in \mathbb{N} : 0 \leq 2i{+}1 < \#ar : ar(2i) * ar(2i{+}1)]$

2. Geef rijtjesexpressies voor
   (a) het rijtje nullen in de rij ar met hun plaats van voorkomen.
   (b) het palindroom dat ontstaat door een rij te spiegelen
   (c) het ritsresultaat van twee even lange rijtjes.

1.2 Verzamelingen

Een eindige verzameling is een rijtje waarin de bewoners (elementen) ten hoogste één keer voorkomen, terwijl de volgorde er niet toe doet. De notatie verschilt een beetje van die van de rijtjes, we enumereren de verzameling door de *elementen* tussen accolades te plaatsen $\{a_0, a_1, \ldots, a_n\}$, De lege verzameling $\{\,\}$ wordt meestal genoteerd als $\emptyset$ en $\{a\}$ staat voor de singleton set waarvan a het enige element is.
Twee verzamelingen zijn gelijk als ze dezelfde elementen hebben.
Behoren tot een verzameling noteren we als $a \in A$ en we zeggen a is een *element* van A. De catenatie van verzamelingen heet *vereniging* en wordt genoteerd als $as \cup bs$. Straks zullen we nog enkele binaire operatoren op verzamelingen zien.
De collectie van alle eindige verzamelingen over het alfabet A , i.e. de eindige deelverzamelingen van A, noteren we als $\mathbb{F}(A)$.

We kunnen net als bij rijtjes nieuwe verzamelingen construeren met behulp van comprehensie. Dat wil zeggen dat we andere verzamelingen, selectie en functietoepassing op hun elementen gebruiken om de elementen van de nieuwe verzameling te beschrijven:

$$dkv(av) \;=\; \{x \;:\; x \in av \;\text{ en }\; x \text{ is oneven} \;:\; x^2\} \tag{3}$$

In het vak Logica en verzamelingenleer zal een vrijwel gelijke notatie gebezigd worden:

$$(\mathsf{set}\ x\ :\ x \in av\ \text{ en }\ x \text{ is oneven}\ :\ x^2)$$

Let op: er zijn géén meervoudige voorkomens. Dus

$$dkv(\{0, 1, 13, -1, 37, 2\}) = \{1, 169, 1369\}\ \text{ en }\ dkr(\emptyset) = \emptyset$$

Merk op dat er verschil is tussen voorkomens en vermeldingen. Er is niet echt iets mis met het meermaals vermelden van een element in de notatie van een verzameling, zoals in de notatie $\{1, 13, 13, 169\}$ voor de verzameling van delers van 169. Het element 13 is dan wel twee keer opgeschreven maar het komt in die verzameling toch maar één keer voor. Dus als we het element 13 er uithalen zit het er niet meer in: dit noteren we als $\{1, 13, 13, 169\} \setminus \{13\} = \{1, 169\}$. (Dit in tegenstelling tot rijtjes waar er toch nog één zou overblijven, maar welke?) De nodeloze herhaling is niet fraai, maar het is niet altijd te vermijden (zie bijvoorbeeld (4)).

1.3 Niet noodzakelijk eindige verzamelingen

We beperken ons niet tot eindige verzamelingen en beschouwen ook niet eindige verzamelingen, zoals de natuurlijke, gehele en reële getallen $\mathbb{N}$, $\mathbb{Z}$ en $\mathbb{R}$. De mathematisch fundamentele aspecten van verzamelingen en niet eindige constructies zullen we hier niet behandelen; we beperken ons tot een eenvoudige schets.
Het zal niet meevallen niet-eindige verzamelingen te enumereren. Soms wordt dan het vertrouwen in de goede wil van de lezer gebruikt (bijvoorbeeld $\{3, 5, 7, \ldots\}$ om de oneven priemen aan te geven). Beter is het niet op de goede wil van de lezer te vertrouwen en comprehensie te gebruiken met selectie en functietoepassing op elementen van andere mogelijk niet eindige verzamelingen:

$$\{n\ :\ n \in \mathbb{N}\ \text{ en }\ n \text{ is priem}\ :\ n\}\ \text{ of verkort }\ \{n \in \mathbb{N} \mid n \text{ is priem}\}$$

Dit priem-voorbeeld geeft ook aan hoe een *deelverzameling* kan worden verkregen door extra eisen aan de elementen te stellen, die we in dat deel willen opnemen met een veel gebruikte verkorte notatie daarvoor.
De notatie voor de *deelverzameling*srelatie tussen twee verzamelingen is:

$$av \subseteq bv \quad \text{en het betekent dat elk element van } av \text{ ook element van } bv \text{ is.}$$

Dus $\{n \in \mathbb{N} | n \text{ is priem}\} \subseteq \mathbb{N}$ en $\{n \in \mathbb{N} | n \text{ is priem en } n \text{ is oneven}\} \subseteq \{n \in \mathbb{N} | n \text{ is priem}\}$. Bovendien gelden trivialiter $\emptyset \subseteq X$ en $X \subseteq X$ voor elke verzameling X, zoöok geldt $\{x\} \subseteq X$ indien $x \in X$.

Nuttige operatoren waarmee we verzamelingen kunnen maken uit andere verzamelingen zijn:

expressie	naam	elementen
$av \cup bv$	vereniging	alle av en bv elementen bijeengeraapt
$av \cap bv$	doorsnede	alle av-elementen die ook in bv voorkomen
$av \setminus bv$	verschil	alle av-elementen die niet in bv voorkomen
$av \times bv$	cartesisch[1] product	tupels (a, b) met $a \in av$ en $b \in bv$
$\mathbb{P}(av)$	machtsverzameling	deelverzamelingen van av

De binaire infix operatoren $\cup, \cap, \setminus$ en $\times$ moeten tussen twee verzamelingen in staan. Dus $A \cup B$ heeft alleen maar zin als A en B verzamelingen zijn, de expressie $A \cup x$ waar x een element van de nieuwe verzameling zou moeten voorstellen is vermoedelijk fout. Hier zal wel $A \cup \{x\}$ bedoeld zijn.

Verzamelingen en elementen horen bij elkaar, "het element zijn" is een relatie tussen twee objecten die qua type bij elkaar moeten passen. Aan de linkerkant staat een ding en aan de rechterkant staat een verzameling van dat soort dingen.
Zo is $13 \in \mathbb{P}(\mathbb{N})$ een dwaze uitspraak, omdat rechts iets staat dat als elementen verzamelingen heeft, terwijl links geen verzameling maar een getal staat.
Ook onzinnig is $\mathit{dertien} \in \{13, 37\}$, want links staat een woord en rechts staat een verzameling getallen.
Daarentegen zijn $\{13\} \in \mathbb{P}(\{13, 37\})$, $[13, 13] \in \mathbb{L}(\{13, 37\})$ en $13 \in \{37\}$ wel zinnig, zij het dat de eerste twee waar zijn en de derde niet.

Van rijtjes kunnen we verzamelingen maken door hun structuur te vergeten, i.e. door ze te ontdoen van hun dubbelen en hun volgorde. Voor een rijtje ar is

$$\mathsf{set}(ar) \;=\; \{a \;:\; a \lessdot ar \;:\; a\} \tag{4}$$

de bijbehorende verzameling, bestaande uit alle rijbewoners van ar. We kunnen ook plezier hebben van de volgorde-notatie van het rijtje door de indicees te gebruiken :

$$\mathsf{set}(ar) \;=\; \{i \;:\; 0 \leq i < l \;:\; ar(i)\} \quad \text{en} \quad \mathsf{set}(\mathsf{rev}(ar)) \;=\; \{i \;:\; 0 \leq i < l \;:\; ar(l-1-i)\}$$

levert dan dus voor deze twee verschillende rijtjes toch dezelfde verzameling op.

Opgave

1. Geef in goed Nederlands weer wat de volgende verzamelingsexpressies voorstellen:
   (a) $\{x, y : x^2 + y^2 = 1 : x\}$
   (b) $X \setminus \{(x, y) \in X \times X : x > y : x\}$
   (c) $\{ar \in \mathbb{L}(A) : \#ar = 5 \;\wedge\; ar \in VD : \mathsf{rev}(ar)\}$, waarbij A het gewone alfabet en VD een woordenboek is.

2. Geef verzamelingsexpressies voor
   (a) alle niet door 13 of 37 deelbare gehele getallen.
   (b) alle gehele getallen die met 481 vermenigvuldigd voorkomen als waarde van een gegeven geheeltallige functie, zeg φ.
   (c) alle deelverzamelingen van $\mathbb{N}$ waar 13 wel maar 37 niet inzit.

[1] Genoemd naar René Descartes een Frans wiskundige en soldaat onder Maurits.

2 Operaties

2.1 Relaties

Het begrip *relatie* dient om een verband tussen sommige elementen van mogelijk verschillende verzamelingen te modelleren. Een adres bijvoorbeeld bestaat uit een naam, een straatnaam, een nummer en een plaatsnaam, mogelijk nog aangevuld met een postcode, telefoonnummers en e-mail adressen. Zo'n adres kan gezien worden als een samengesteld element van een fiks cartesisch product, maar het kan ook gezien worden als een collectie elementen, die samen een relatie hebben, de "adres-relatie". Lang niet alle combinaties van naam, straat, nummer en plaats hebben die adres-relatie met elkaar, maar in een redelijke database kunnen er toch erg veel van die adressen zitten. Alle combinaties die een adres-relatie tot elkaar hebben vormen dan samen een relatie, zeg *adres* en een combinatie

$$a = \{\mathsf{naam} \mapsto \mathit{arie\ aarts}, \mathsf{straat} \mapsto \mathit{akkerweg}, \mathsf{nummer} \mapsto 8, \mathsf{plaats} \mapsto \mathit{andijk}\}$$

heeft een adres-relatie als $a \in \mathit{adres}$.

In het algemeen zouden we een relatie kunnen definiëren als een deelverzameling van een cartesisch product, waarbij de componenten benoemd kunnen worden zoals in het voorbeeld of waarbij de gerelateerde elementen in een tupel geschreven kunnen worden, als in

$$(\mathit{arie\ aarts}, \mathit{akkerweg}, 8, \mathit{andijk}) \in \mathit{adres} \subseteq \mathsf{Naam} \times \mathsf{Straat} \times \mathbb{N} \times \mathsf{Plaats}$$

In het bovenstaande zijn Naam, Straat en Plaats de verzamelingen waaruit de namen, straatnamen en de plaatsnamen worden geput.

We zullen in het vervolg alleen relaties tussen elementen van twee verzamelingen bekijken, de zogenaamde *binaire relaties*. Een (binaire) relatie $R : X \sim Y$is een deelverzameling R van $X \times Y$. Twee elementen $x \in X$ en $y \in Y$ zijn gerelateerd volgens R als $(x, y) \in R$, we zullen dat ook wel noteren als $x\ R\ y$.

Voorbeelden van relaties zijn

typering	naam	voorbeeld
$< \ : \ \mathbb{R} \sim \mathbb{R}$	kleiner dan	$13{,}37 < \sqrt{481}$
$\in \ : \ A \sim \mathbb{P}(A)$	element van	$13 \in \{1, 13, 37, 481\}$
$\Subset \ : \ A \sim \mathbb{L}(A)$	bewoner van	$37 \Subset [1, 13, 1, 37, 37, 1, 481]$
$\subseteq \ : \ \mathbb{P}(A) \sim \mathbb{P}(A)$	bevat in	$\{13, 37\} \subseteq \{1, 13, 37, 481\}$
$\preceq \ : \ \mathbb{L}(A) \sim \mathbb{L}(A)$	prefix van	$[1, 13, 1] \preceq [1, 13, 1, 37, 37, 1, 481]$
$\mid \ : \ \mathbb{N} \sim \mathbb{N}$	is deler van	$13 \mid 481$

Een andere bekende maar ook lastige relatie is de gelijkheid:

$= \ : \ A \sim A$ betekenend dat a en b gerelateerd zijn, $a = b$, als ze hetzelfde zijn

Wat dan "hetzelfde" weer betekent zullen we moeten afspreken. (Is $2 + 2$ hetzelfde als 4? Dat hangt er van af of we de expressie bedoelen of de bijbehorende waarde na de optelling.)

Andere populaire relaties in besprekingen van het relatie-begrip zijn die van ouderschap en partnerschap en collegadom, die mensen uit een tevoren gegeven collectie aan elkaar relateren. Laten we ze $O, P, C : \mathsf{Mens} \sim \mathsf{Mens}$, met de betekenis dat $x\ O\ y$ aangeeft dat x ouder is van y en iets dergelijks, maar symmetrisch geldt voor P en C. Die symmetrie is het gevolg van de verwoording van de relatie, de begrippen partner en collega zijn in onze taal nu een maal symmetrisch. Hoe zien we dat nu aan de relatie?
Daarvoor moeten we kijken naar de inverse relatie. De relatie O is niet symmetrisch, het linker element in $x\ O\ y$ heeft een andere rol dan het rechter. In het bijzonder zal niet ook gelden dat $y\ O\ x$, dat zou een rare wereld worden. Als we willen uitdrukken dat y een kind is van x, kunnen de relatie "omdraaien", we schrijven dan $y\ O^{\leftarrow} x$.

In het algemeen:
Laat R een binaire relatie zijn, dan is de *inverse* $R^{\leftarrow}$ van R gedefiniëerd door

$$p\ R^{\leftarrow} q \quad \text{dan en slechts dan als} \quad q\ R\ p$$

Een relatie is *symmetrisch* als $R\ =\ R^{\leftarrow}$.

Als we nu willen uitdrukken dat *arie* een collega is van de partner van *alie* dan zouden we die partner eventjes kunnen aangeven met x en noteren dat *arie* C x en x P *alie*. Deze doorschakeling of compositie van relaties wordt aangeduid met een ";" resulterend in *arie* $C; P$ *alie*.

In het algemeen:
Laat R en S binaire relaties zijn dan wordt de *relatiecompositie* $R; S$ van R en S gedefiniëerd door

$$x\ R; S\ z \quad \text{dan en slechts dan als er een } y \text{ is zo, dat} \quad x\ R\ y \text{ en } y\ S\ z$$

We kunnen de begrippen inverse en compositie natuurlijk ook combineren en operaties uit de verzamelingenleer daarbij gebruiken. Zo worden "siblings" (zoiets als nestgenoten, een echt Nederlands woord daarvoor ken ik niet) die ook nog elkaars collega zijn gegeven door $(O^{\leftarrow};\ O)\ \cap\ C$

Opgave

1. Zij $\Delta = \{x : \mathsf{true} : (x, x)\}$ de *identiteits*relatie of *diagonal*. Wat stellen de volgende relatie expressies voor?
   (a) $(P; C; O)\ \cap\ \Delta$
   (b) $P; P\ \subseteq\ \Delta$
   (c) $\{x, y : x\ (O; O; P)\ y : x\}$

2. Geef relatie expressies voor
   (a) Iedereen heeft een partner.
   (b) Ouders van collegae zijn collegae.
   (c) De ouders van mensen die een collega als partner hebben.

2.2 Functies

Het begrip functie is bekend van de middelbare school, hoewel daar niet altijd onderscheid wordt gemaakt tussen een functie en haar beschrijving, terwijl er daardoor weer een lastig verschil tussen een functie en haar grafiek ontstaat. Wat volgt is de versie van het begrip functie, zoals dat in het vervolg van de opleiding het meest zal worden gebruikt.
Een functie is een speciaal soort relatie die elementen van twee verzamelingen op een zodanig "nette" wijze aan elkaar relateert, dat over argument en beeld gesproken kan worden. Er is een domein waaruit de argumenten komen en een codomein waarin de beeldelementen zitten. Kenmerkend voor een functie is dat er per argument ten hoogste één beeld is (en meestal precies één).
Het is te doen gebruikelijk om de typering, de vermelding van domein en codomein, mee te geven met elke definitie van een functie. We schrijven derhalve

$$f : A \longrightarrow B \quad \text{en} \quad f(a) = b \quad (\text{of zoals bij relaties} \;\; (a,b) \in f \;\; \text{maar niet} \;\; a \; f \; b)$$

De functie heet f (en dus niet $f(a)$), het domein van f is A (het definitiegebied van f is daar weer een deelverzameling van) en het codomein van f is B (de beeldverzameling van f is daarvan een deelverzameling).

Eenvoudige functies die we ook direct als zodanig herkennen zijn bijvoorbeeld kwadratering, lengteberekening van rijtjes en adresberekening op grond van postcode en huisnummer. Sommige functies zijn ons zo vertrouwd, dat we ze haast niet meer als functies herkennen, zoals de optelling ($+$) en de vermenigvuldiging ($*$). De bijbehorende functie-applicaties noteren we meestal door het functiesymbool tussen de argumenten in te zetten (infix notatie) in plaats van ervoor (prefix notatie), zoals in

$$+ : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N} \quad \text{waar } m+n \text{ het resultaat is van de functie } + \text{ op } m \text{ en } n$$

Niet alleen is iedere functie een relatie (met de speciale uniciteitseis op het beeld), iedere relatie kan als functie worden gezien. Zij $R : X \sim Y$, dan kunnen we een bijbehorende functie $\overline{R}$ definiëren door:

$$\overline{R} \;:\; X \times Y \longrightarrow \mathbb{B} \;\; \text{met} \;\; \overline{R}(x,y) = \mathsf{true} \text{ dan en slechts dan als } x \; R \; y$$

Het type $\mathbb{B}$ wordt gebruikt om de waarden "waar" en "onwaar" aan te duiden, [2] het type heeft dus slechts twee elementen die we aangeven met de symbolen true en false, waarbij de eventuele gelijkenis met Engelse woorden niet geheel toevallig is.
We zullen het onderscheid tussen R en $\overline{R}$ niet echt maken en altijd R schrijven, of we nu de relatie of de boolean-waardige functie bedoelen. Zoals in

$$\begin{array}{rcll} \leqslant & : & A \times \mathbb{L}(A) \longrightarrow \mathbb{B} & \text{bewoner van} \\ \in & : & A \times \mathbb{P}(A) \longrightarrow \mathbb{B} & \text{element van} \\ \subseteq & : & \mathbb{P}(A) \times \mathbb{P}(A) \longrightarrow \mathbb{B} & \text{bevat in} \end{array}$$

[2] De $\mathbb{B}$ staat voor "booleans". Het type van de booleans is genoemd naar George Boole, een wiskundige uit de negentiende eeuw die aan de wieg van de symbolische logica heeft gestaan.

Andere inmiddels min of meer bekende functies zijn:

$\#$	$: \mathbb{L}(A) \to \mathbb{N}$	aantal bewoners van rijtjes
$\#$	$: \mathbb{F}(A) \to \mathbb{N}$	aantal elementen (dus alleen voor eindige verzamelingen)
$+\!\!+$	$: \mathbb{L}(A) \times \mathbb{L}(A) \longrightarrow \mathbb{L}(A)$	catenatie
$\cup\ ,\ \cap\ ,\ \setminus$	$: \mathbb{P}(A) \times \mathbb{P}(A) \longrightarrow \mathbb{P}(A)$	vereniging, doorsnede, verschil

Het feit dat de functie f in argument a de waarde b heeft kan worden gebruikt als middel om een functie vast te leggen. Vaak, namelijk, worden functies gedefinieerd door in plaats van b een expressie te geven in termen van het argument a, en/of door constructies met andere bekende functies te bouwen. We hebben er al een paar gezien, namelijk

$$\begin{array}{rcll} dkr & : & \mathbb{L}(\mathbb{Z}) \longrightarrow \mathbb{L}(\mathbb{Z}) & \text{in} \quad (1) \\ \mathsf{rev} & : & \mathbb{L}(A) \longrightarrow \mathbb{L}(A) & \text{in} \quad (2) \\ dkv & : & \mathbb{P}(\mathbb{Z}) \longrightarrow \mathbb{P}(\mathbb{Z}) & \text{in} \quad (3) \\ \mathsf{set} & : & \mathbb{L}(A) \longrightarrow \mathbb{F}(A) & \text{in} \quad (4) \end{array}$$

Andere voorbeelden zijn:

typering	definitie
$0^\bullet : \mathbb{N} \longrightarrow \mathbb{N}$	$0^\bullet(n) = 0$
$\mathsf{succ} : \mathbb{N} \longrightarrow \mathbb{N}$	$\mathsf{succ}(n) = n + 1$
$[\cdot] : A \to \mathbb{L}(A)$	$[\cdot](a) = [a]$
$\pi_2 : A \times B \to B$	$\pi_2((a, b)) = b$
$\uparrow : \mathbb{Z} \times \mathbb{Z} \longrightarrow \mathbb{Z}$	$p \uparrow q = \textbf{if}\ p \leq q \to q\ []\ p \geq q \to p\ \textbf{fi}$
$\wedge : \mathbb{B} \times \mathbb{B} \longrightarrow \mathbb{B}$	$a \wedge b = \textbf{if}\ a = b \to a\ []\ a \neq b \to \mathsf{false}\ \textbf{fi}$ [3]
$\vee : \mathbb{B} \times \mathbb{B} \longrightarrow \mathbb{B}$	$a \vee b = \textbf{if}\ a = b \to a\ []\ a \neq b \to \mathsf{true}\ \textbf{fi}$ [4]
$\neg : \mathbb{B} \longrightarrow \mathbb{B}$	$\neg(\mathsf{true}) = \mathsf{false}$ en $\neg(\mathsf{false}) = \mathsf{true}$
$\Rightarrow : \mathbb{B} \times \mathbb{B} \longrightarrow \mathbb{B}$	$a \Rightarrow b = \textbf{if}\ a = b \to \mathsf{true}\ []\ a \neq b \to \neg(a)\ \textbf{fi}$ [5]
$\equiv : \mathbb{B} \times \mathbb{B} \longrightarrow \mathbb{B}$	$a \equiv b = \textbf{if}\ a = b \to \mathsf{true}\ []\ a \neq b \to \mathsf{false}\ \textbf{fi}$ [6]
$\mathsf{fib} : \mathbb{N} \longrightarrow \mathbb{N}$	$\mathsf{fib}(n) = \textbf{if}\ n \leq 1 \to n\ []\ n \geq 2 \to \mathsf{fib}(n-1) + \mathsf{fib}(n-2)\ \textbf{fi}$
$\sigma\alpha : \mathbb{L}(\mathbb{Z}) \longrightarrow \mathbb{Z} \times \mathbb{N}$	$\sigma\alpha(r) = \textbf{if}\ r = [\,] \to (0,0)\ []\ r = [b] +\!\!+ s \to (b,1) + \sigma\alpha(s)\ \textbf{fi}$ [7]
$\mathsf{avg} : \mathbb{L}(\mathbb{Z}) \longrightarrow \mathbb{Z}$	$\mathsf{avg}(r) = \pi_1(\sigma\alpha(r))/\pi_2(\sigma\alpha(r))$ mits $r \neq [\,]$

Merk op dat de definiërende expressies voor fib en $\sigma\alpha$ refereren aan diezelfde functies. Dit lijkt een cirkelredenering. We moeten inderdaad voorzichtig zijn met dit soort *recursieve* definities. Voor het moment nemen we genoegen met een definitie waarbij het duidelijk is dat er per argument slechts een eindige reeks "zelf"referenties zal optreden.

[3] Equivalent is: $a \wedge b = \mathsf{true}$ dan en slechts dan als beide argumenten dat zijn.
[4] Equivalent is: $a \vee b = \mathsf{true}$ dan en slechts dan als ten minste één der twee argumenten dat is.
[5] Equivalent is: $a \Rightarrow b = \neg(a) \vee b$, i.e. $a \Rightarrow b = \mathsf{true}$ als a niet geldt of b wel.
[6] Equivalent is: $a \equiv b = (a \Rightarrow b) \ \vee\ (a \Leftarrow b)$, i.e. als beide argumenten dezelfde waarde hebben.
[7] De optelling gaat hier coördinaatsgewijs en had wellicht genoteerd moeten worden als $(+,+)$.

Behalve definiëren en uitrekenen kun je met functies ook componeren. Als je op een argument eerst functie f wilt toepassen en daarna op het resultaat functie g wilt loslaten, dan kan dat natuurlijk alleen als het resultaat van f ook daadwerkelijk door g als argument kan worden geaccepteerd. Als $f : A \longrightarrow B$ en $g : B \longrightarrow C$ functies zijn waarbij het codomein van f overeenkomt met het domein van g, dan wordt de *compositie* van g na f gegeven door

$$g \circ f \ : \ A \longrightarrow C \quad \text{en} \quad (g \circ f)(a) \ = \ g(f(a))$$

De eis op het overeenkomen van codomein van f en domein van g wordt meestal wat verzacht, in de praktijk vinden we de compositie al acceptabel als het beeld van f bevat is in het definitiegebied van g.

Specificeren en programmeren komt veelal neer op het vinden, construeren of berekenen van geschikte functies. Meestal worden die via compositie opgebouwd uit andere. Het opdelen van een probleem zo, dat het resultaat voor het geheel kan worden verkregen door compositie van de resultaten van de delen is een belangrijke "techniek" in de informatica.

Net als bij relaties kunnen we ook de inverse van functies bekijken. De inverse van een functie is niet noodzakelijk weer een functie, erger nog, meestal is het geen functie. Zodra er twee argumenten zijn waarvoor de functie een zelfde waarde geeft is de inverse al geen functie meer.
Zij $f : X \longrightarrow Y$ een functie, dan is de *inverse* $f^{\leftarrow}$ van f een relatie

$$f^{\leftarrow} : Y \sim X \ \text{ gedefiniëerd door } \ y \, f^{\leftarrow} x \ \equiv \ f(x) = y.$$

We schrijven in dit geval ook wel $(y, x) \in f^{\leftarrow}$ of $x \in f^{\leftarrow}(y)$.

Opgave

1. Beschrijf wat de volgende functies berekenen:
   (a) $f : \mathbb{L}(A) \times \mathbb{L}(A) \to \mathbb{L}(\mathbb{L}(A))$ gegeven door

   $$f((ar, br)) = [i : 0 \leq i < \#ar \downarrow \#br : [ar(i), br(i)]]$$

   (b) $g : \mathbb{L}(\mathbb{L}(A)) \to \mathbb{L}(A)$ gegeven door

   $$g([\,]) = [\,], \quad g([a]) = a \ \text{ en } \ g(ar +\!\!+ br) = g(ar) +\!\!+ g(br)$$

   (c) $g \circ f$ met f, g als hierboven.

2. Geef definities van functies die het volgende berekenen
   (a) De verzameling grootouders van mensen (gegeven de ouder relatie O).
   (b) De som van alle delers van een natuurlijk getal.
   (c) Eindige machtreeksen van een gegeven getal z, informeel: $f(n) = z^0 + ... + z^n$.

2.3 Expressies

De expressies zoals we die in functiedefinities willen gebruiken moeten natuurlijk netjes getypeerd zijn. Bovendien willen we de variabelen die daarin voorkomen ook goed kunnen onderscheiden van de andere waarden; die variabelen kunnen we het beste altijd zien als argumenten van de functie die met de betrokken expressie wordt gedefinieerd.

We zullen hier nog een paar voorbeelden van expressies en hun notatie geven die betrekking hebben op rijtjes en verzamelingen, de bulktypen. Vooral denken we dan aan expressies die uitdrukken dat we met alle dingen in een "bulk" iets gelijksoortigs doen of dat we iets herhalen voor alle leden van de bulk. De notatie daarvoor zal veel weg hebben van de notatie zoals we die al gebruikt hebben in de expressies in (1, 3). Voorbeeld: sommering van rijen of van verzamelingen, zij dus $ar \in \mathbb{L}(\mathbb{Z})$ en $av \in \mathbb{F}(\mathbb{Z})$ dan zijn

$$(+\, a \,:\, a \lessdot ar \,:\, a^2) \quad \text{en} \quad (+\, x \,:\, x \in av \,:\, x^2)$$

de sommen van de kwadraten van de bewoners van de bulks.
We zien ook hier de driedeling: declaratie van de dummies, definitie van het domein van de dummies en de bepaling van de te behandelen term. Nu echter komt er een operator bij, die vertelt wat we met de termen gaan doen. In dit geval is dat dus optellen, $+ : \mathbb{Z} \times \mathbb{Z} \longrightarrow \mathbb{Z}$, dat herhaald wordt uitgevoerd. Voor de rij ar doen we dat dus met alle rij-bewoners en voor de verzameling av doen we dat met alle elementen.
We noemen die operator ook wel de *quantor* en de expressie een *quantificatie*. Uit mathematisch-traditionele overwegingen wordt die +-quantor ook wel geschreven als Σ.
Behalve de optelling kunnen we ook als quantor-operator op $\mathbb{Z}$ de vermenigvuldiging $(*)$, maximum $(\uparrow)$ en minimum $(\downarrow)$ kiezen om expressies met waarden in $\mathbb{Z}$ te construeren.
Enige voorzichtigheid is geboden bij dit soort expressies voor niet eindige domeinen. Wat zou, bijvoorbeeld, $(+x \,:\, x \in \mathbb{N} \,:\, (-1)^x * x)$ moeten betekenen?

Voor quantificaties over rijtjes van rijtjes kunnen we de catenatie $+\!\!+$ gebruiken, zoals in het volgende voorbeeld. Laat daartoe $arr \in \mathbb{L}(\mathbb{L}(A))$ een rijtje van rijtjes zijn:

$$(+\!\!+ \;\; r \,:\, r \lessdot arr \;\; \text{en} \;\; \#r \;\text{is even} \,:\, r +\!\!+ r)$$

alle rijtjes van even lengte worden verdubbeld en aan elkaar geplakt tot een nieuw rijtje.

Voor quantificaties van bulks van verzamelingen hebben we de operatoren $\cup$ en $\cap$. Laat $avr \in \mathbb{L}(\mathbb{P}(A))$ en $avv \in \mathbb{P}(\mathbb{P}(A))$ een rijtje verzamelingen en een verzameling verzamelingen zijn, dan berekenen

$$(\cup\, r \,:\, r \lessdot avr \,:\, r) \quad \text{en} \quad (\cap\, v \,:\, v \in avv \,:\, v)$$

respectievelijk de verzameling van alle elementen die in enige bewoner van avr zitten en de verzameling van elementen die in elke bewoner van avv zitten.

Quantificaties van waarheidswaarden (elementen van $\mathbb{B}$) maken we met behulp van de operatoren $\vee$ en $\wedge$. Zoals in

$$(\wedge\, i \,:\, 0 \leq i < \#ar \,:\, i < ar(i)) \;\; \text{en} \;\; (\vee\, x \,:\, x \in av \,:\, x^2 \in av)$$

waarin berekend wordt of elk rij-element van ar groter is dan zijn index en of er een element in av voorkomt waarvan het kwadraat ook in av zit.

Merk op dat het type van de expressie wordt bepaald door het type van de quantor. Als de quantor, bijvoorbeeld, een $+\!\!+$ is dan is het resultaat een rijtje, is het een $\cup$ dan is het resultaat een verzameling.
De operatoren die als quantor zijn toegestaan moeten aan zekere algebraïsche eisen voldoen. Daarop zullen we hier niet ingaan, behalve dan dat we wijzen op de speciale rol van het lege domein. Omdat het lege domein overal aan kan worden toegevoegd, mag de waarde door die toevoeging niet veranderen. Daarom kiezen we in die gevallen voor een waarde die "neutraal" is. Voor $+$ en $*$ zijn dat 0 en 1, voor de catenatie $+\!\!+$ is dat $[\,]$, voor $\cup$ is dat $\emptyset$ en voor $\wedge$ en $\vee$ zijn dat true respectievelijk false. Zo geldt, bijvoorbeeld,

$$(+\,i \;:\; i \in \emptyset \;:\; h(i)) \;=\; 0, \quad \text{en} \quad (+\!\!+\, b \;:\; b \lessdot [\,] \;:\; \varphi(b)) \;=\; [\,]$$

Merk echter op dat er ook operaties zijn die niet zondermeer een neutraal element hebben. Het is niet a priori duidelijk (tenzij goed afgesproken) wat de neutrale elementen zijn van $\uparrow$, $\downarrow$ en $\cap$.

Opgave
Laat arr een rij van rijtjes en ars een verzameling van rijtjes van gehele getallen zijn, en zij ar een rijtje gehele getallen.

1. Beschrijf wat de volgende expressies betekenen,
   (a) $[ar : ar \lessdot arr : (\uparrow a : a \lessdot ar : a)]$
   (b) $(\downarrow p, q : 0 \leq p \leq q \leq \#ar \;\wedge\; (+i : p \leq i < q : ar(i)) \geq 13 : q - p)$
   (c) $(+i : i \in rs : i)/\#rs$, waarbij $rs = \{ar : ar \in ars : (+a : a \lessdot ar : a)\}$.

2. Geef expressies voor de volgende waarden
   (a) Maximale lengte van enig segment (consecutieve deelrij) van ar met louter nullen.
   (b) Vereniging van alle delers van alle rij-elementen van alle rij-elementen van arr.
   (c) Gemiddelde rijsom van ars.

3 Predikaten

Functies en expressies met waarden in $\mathbb{B}$ nemen een prominente plaats in in specificaties; daarmee worden de condities uitgedrukt waaraan moet worden voldaan (bijvoorbeeld constraints en invarianten).
Eigenlijk is een $\mathbb{B}$-waardige functie hetzelfde als een verzameling, bij een functie $f : A \longrightarrow \mathbb{B}$ hoort een verzameling

$$F = \{a \in A \mid f(a) = \mathsf{true}\}$$

en andersom definieert een deelverzameling $G \subseteq A$ een functie $g : A \longrightarrow \mathbb{B}$ door

(5) $\qquad g(a) = \textbf{if } a \in G \rightarrow \mathsf{true} \;[]\; a \notin G \rightarrow \mathsf{false}\; \textbf{fi} \quad$ of liever $\quad g(a) = (a \in G)$.

Op deze wijze kunnen we met behulp van $\mathbb{B}$-waardige functies allerlei deelverzamelingen (of condities) aangeven. We noemen deze functies ook wel *predikaten*. De argumenten van de bijbehorende $\mathbb{B}$-waardige functie noemen we de *variabelen* van het predikaat.

- Zo is $x < y$ een predikaat in x en y, dat voor $x = 0$ en $y = 0$ onwaar is (de waarde false heeft) en voor $x = 13$ en $y = 37$ waar is.
- De expressie $p \in A$ is een predikaat in p en A, vaak echter wordt van te voren een aantal variabelen gefixeerd. Bijvoorbeeld, als alles binnen A gebeurt, zou A als constante kunnen worden beschouwd en wordt daarmee $p \in A$ een predikaat in de variabele p (zie bijvoorbeeld de definitie in (5)).
- De gelijkheid $13 * 37 = 481$ is een predikaat zonder variabelen (een *propositie*), die met de gebruikelijke vermenigvuldiging waar is (waarde true heeft).

Twee zeer belangrijke predikaatvormen hebben we al gezien in de vorige sectie: predikaten die quantificaties zijn met als operatoren $\vee$ of $\wedge$. Laat P en Q twee predikaten zijn in x, dan zijn dat

$(\vee\, x \,:\, P(x) \,:\, Q(x))$ ook genoteerd als $\exists x \cdot (P(x) \wedge Q(x))$ of $(\exists\, x \,:\, P(x) \,:\, Q(x))$
$(\wedge\, x \,:\, P(x) \,:\, Q(x))$ ook genoteerd als $\forall x \cdot (P(x) \Rightarrow Q(x))$ of $(\forall\, x \,:\, P(x) \,:\, Q(x))$

We noemen ze achtereenvolgens de *existentiële* quantificatie en de *universele* quantificatie. In het vak Logica en verzamelingenleer wordt een bijna gelijke notatie gebruikt:

$$\exists_x[\, P(x) \,:\, Q(x)\,] \quad \text{en} \quad \forall_x[\, P(x) \,:\, Q(x)\,]$$

De woordelijke betekenis is achtereenvolgens:
"Er bestaat een x die aan P voldoet zo, dat daarvoor Q waar is."
"Voor alle x waarvoor P waar is is Q waar."

Deze predikaatvormen kunnen ook nog worden gecombineerd, zoals te zien is in:

$$(\forall\, x \in \mathbb{N} \,:\, x \text{ is priem} \,:\, (\exists\, y \in \mathbb{N} \,:\, y > x \,:\, y \text{ is priem}))$$
$$(\forall\, x, y \in \mathbb{Z} \,:\, (\exists\, z \in \mathbb{Z} \,:\, 0 \leq z < 5 \,:\, x^2 + y^2 = z^2) \,:\, x * y = 0)$$

Deze predicaten zijn beide waar, maar door in plaats van $\mathbb{N}$ en $\mathbb{Z}$ andere verzamelingen te kiezen wordt dat mogelijk geheel anders.

Opgave

1. Geef de betekenis van de volgende predikaten weer als een Nederlandse zin.
   (a) $(\forall x, y, z : xRy \ \wedge\ xRz : y = z)$
   (b) $(\exists a, b, c : a, b, c \in \mathbb{Z} : a^n + b^n = c^n) \Rightarrow n \leq 2$
   (c) $(\uparrow m : m \in M : m)$ bestaat $\equiv (\exists n : n \in \mathbb{N} : n{+}1 = \#M)$
2. Geef predikaten voor
   (a) Een gegeven woord is een palindroom dat ten hoogste 13 letters lang is.
   (b) Als een gegeven woord een palindroom bevat dan heeft dat hoogstens 13 letters.
   (c) Van een gegeven woord is geen enkel segment met lengte ten minste 14 een palindroom.

4 Voorbeeldje

Voor een badmintonclub die haar leden uitslagen laat produceren via enkelspel, dubbelspel en mix partijtjes moet een informatiesysteem gemaakt worden om de administratie van de leden L en de uitslagen U bij te houden en om antwoorden te genereren op verschillende vragen over die leden en de uitslagen. Hoe die leden en die uitslagen er nu precies uit zien doet er nog even niet toe, dat zou (deels) kunnen volgen uit de informatie die we er van willen gebruiken. Wel is het te verwachten dat we willen weten wie een partijtje gewonnen of verloren heeft en van welke kunne een lid is. Vooralsnog zullen we het louter met deze informatie doen.

Laat de typen Uitslag en Lid gegeven zijn waaruit de uitslagen en de leden worden gerekruteerd. Dan zijn

$$L \subseteq \mathsf{Lid} \quad \text{en} \quad U \subseteq \mathsf{Uitslag}$$

en we hebben de beschikking over de functies

$$k : \mathsf{Lid} \longrightarrow \{♂, ♀\} \quad \text{en} \quad v, w : \mathsf{Uitslag} \longrightarrow \mathbb{P}(\mathsf{Lid})$$

die de kunne van de leden bepaalt en de verliezers en winnaars van de partijtjes.
Merk op dat de verliezers en winnaars in een verzameling gezet worden, in het dubbelspel en in de mix zijn het er tenslotte telkens twee.

Met deze basisgegevens kunnen we hulpverzamelingen, hulpfuncties en predikaten maken, zoals

$$M = \{\ l\ :\ l \in L\ :\ k(l) = ♂\ \} \quad \text{en} \quad V = \{\ l\ :\ l \in L\ :\ k(l) = ♀\ \}$$
$$\mathit{punt} : L \longrightarrow \mathbb{N} \quad \text{waarbij} \quad \mathit{punt}(l) = 2 * \#\{u \in U \mid l \in w(u)\}$$
$$(\forall u\ :\ u \in U\ :\ v(u) \cap w(u) = \emptyset \quad \wedge \quad \#v(u) = \#w(u)\ ) \tag{6}$$

We zullen vanaf nu eisen dat de uitslagen voldoen aan de conditie (6), die uitdrukt dat de partijtjes tussen gelijke aantallen leden gaan die zich elk maar aan één kant van het net mogen bevinden.

Als we in het informatiesysteem van de badmintonclub een applicatie willen hebben die de ranglijst van haar manlijke leden in het enkelspel oplevert, dan kunnen we het resultaat specificeren als:

$$R \in \mathbb{L}(M) \quad \text{zo, dat} \quad (\forall i\ :\ 0 < i < \#R\ :\ \mathit{puntenkel}(R(i-1)) \geq \mathit{puntenkel}(R(i))\ )$$

waarbij de functie *puntenkel* een variatie is op *punt* en gegeven wordt door

$$\mathit{puntenkel} : L \longrightarrow \mathbb{N} \quad \text{en} \quad \mathit{puntenkel}(l) = 2 * \#\{u \in U \mid \{l\} = w(u)\}$$

Maar dit is nog niet alles! Als we een lijst krijgen waarop slechts een lid vele malen voorkomt mogen we niet klagen. Dat voldoet ook aan de specificatie. We zullen moeten

garanderen dat elk lid precies één keer in de lijst voorkomt. Dat kunnen we doen door ook nog te eisen dat $\mathsf{set}(R) = M \ \wedge \ \#R = \#M$.

Mochten we binnen die ranglijst zelf geïnteresseerd zijn in het aantal behaalde punten dan kunnen we dat specificeren als

$$RP \in \mathbb{L}(M \times \mathbb{N}) \quad \text{en} \quad (\forall i \ : 0 \leq i < \#R \ : \ RP(i) = (R(i), \mathit{puntenkel}(R(i))) \ )$$

$$\text{en} \quad \mathsf{set}[rp : rp \lhd RP : \pi_1(rp)] = G \ \wedge \ \#RP = \#M$$

Opgave

1. Geef de betekenis van de volgende expressies weer als een Nederlandse zin.
   (a) $\{k \in V, l \in M, u \in U \ : \ v(u) = \{l\} \wedge w(u) = \{k\} \ : \ k\}$
   (b) $(\exists k \in V, l \in M, u \in U \ : \ k \in w(u) \wedge l \in v(u) \ : \ (\forall u' \in U : \#w(u') = 1 : l \notin v(u')))$
   (c) $(\forall k \in V \ :: \ \#(P(k) \cap M) \leq 1) \ \wedge \ (\forall l \in M \ :: \ \#(P(l) \cap V) \leq 1)$, waarbij $P(x) = (\cup u : x \in v(u) : v(u)) \cup (\cup u : x \in w(u) : w(u))$.

2. Geef expressies voor
   (a) Alle winnaars hebben ook ooit eens verloren.
   (b) De leden die met de meeste andere leden samengespeeld hebben.
   (c) De leden hebben niet aan partnerruil gedaan.