Module 3 Functies, getallen

Onderwerp	Definitie van functies van één of meer variabelen, rationale, reële en complexe getallen.
Voorkennis	VWO-stof over functies, complexe getallen.
Expressies	`->`, `eval`, `unapply`, `@`, `piecewise`, `evalf`, `evalc`, `Digits`, `round`, `trunc`, `frac`, `floor`, `ceil`, `convert(rational)`, `sin`, `cos`, `tan`, `arcsin`, `arccos`, `arctan`, `exp`, `^`, `I`, `Pi`, `infinity`, `inifcn`, `ininame`, `plot`, `abs`, `ln`, `log`, `signum`, `csgn`, `Re`, `Im`, `argument`, `conjugate`, `polar`, `sqrt`, `root`, `surd`, `RootOf`, `allvalues`
Bibliotheken	`RealDomain`
Zie ook	Module 5, 7, 28

3.1 Functies, complexe getallen

Functies. Laat A en B verzamelingen zijn. Een afbeelding (of *functie*) f van A naar B is een voorschrift dat aangeeft hoe aan *elk* element x van A *één* element $f(x)$ van B wordt toegevoegd. We noteren dit vaak als $f : x \mapsto f(x)$. De verzamelingen A en B noemen we respectievelijk het *domein* en *bereik* van f. Het symbool f is de *naam* van de functie, $f(x)$ is de functie*waarde* die hoort bij het *argument* x. Wanneer A en B deelverzamelingen zijn van $\mathbb{C}$ spreken we van een *complexe* functie.

Complexe getallen. Een *complex getal* z kan worden geschreven als $z = x + iy$, met $x, y \in \mathbb{R}$. Dit is de *Cartesische vorm* van een complex getal. Hierin is x het reële deel van z, aangegeven met $\mathrm{Re}(z)$ of $\Re(z)$ en y het imaginaire deel, aangegeven met $\mathrm{Im}(z)$ of $\Im(z)$.
De *geconjugeerde* (of *complex toegevoegde*) wordt aangegeven met $\overline{z}$ en is gedefinieerd als $\overline{z} = x - iy$ als $z = x + iy$.
De *polaire vorm* van een complex getal is $z = re^{i\varphi}$, met $0 \leq r$ en $\varphi \in \mathbb{R}$. Er geldt dat

$$re^{i\varphi} = r\cos\varphi + r\,i\sin\varphi\,.$$

We noemen $r = |z|$ de *absolute waarde* van z en φ het *argument*[7] van z. Omdat $re^{i\varphi} = re^{i\varphi + 2k\pi i}$ voor alle $k \in \mathbb{Z}$, is de waarde van φ niet

[7] Merk op dat het woord 'argument' twee volstrekt verschillende betekenissen kan hebben, namelijk de x in $f(x)$, of de φ in $re^{i\varphi}$.

eenduidig bepaald. De φ met $-\pi < \varphi \leq \pi$ heet de *hoofdwaarde* van het argument.

3.2 Functies in Maple

Functies. De eenvoudigste manier om in Maple een functie te definiëren is met de operator `->`. (Er zijn ook nog andere manieren, daar gaan we in Module 7 en 28 verder op in.)

`->`

Bijvoorbeeld, als we willen dat f de functie $x \mapsto 2x$ voorstelt, dan moeten we f in Maple definiëren als `f := x -> 2*x`. In de regel is het niet mogelijk het domein en het bereik van f in het voorschrift mee te geven.

De `x` die in de uitdrukking `f := x -> 2*x` wordt gebruikt is een *dummy*; met `f := y -> 2*y` zouden we precies dezelfde functie hebben gedefinieerd.

`f( )`
`apply`

We kunnen de functie f toepassen (Engels: *apply*) op een argument a met het commando `f(a);`. Een alternatief voor de aanroep `f(a)` is daarom: `apply(f,a)`.

Een functie hoeft geen naam te hebben. Maple heeft geen probleem met een opdracht als `(x -> 2*x)(t);`, het resultaat is, zoals verwacht, $2t$.

`eval`

Expressies. Andersom, als `F := x^2` dan is `F` *geen* functie. Om de waarde van `F` voor $x = 2$ uit te rekenen is een substitutie- of evaluatie-opdracht (`subs` of `eval`) nodig, bijvoorbeeld: `eval(F, x=2);`. Zie verder §4.5. We spreken in dat geval van `F` als *expressie*, ter onderscheiding van de *functie* `f`.[8] Als `f` een functie is, dan is `f(a)` een expressie.

`unapply`

Expressie omzetten in een functie. Men kan van de expressie `F` een functie maken met de opdracht `f := unapply(F,x);`[9]. Een functie van twee variabelen maakt men eenvoudig met `unapply(F,x,y)` als `F` een expressie is waarin de variabelen `x` en `y` voorkomen.

Dus: met `apply` maakt men van een functie een expressie; met `unapply` maakt men van een expressie een functie.

[8] Eigenlijk zouden we f een *procedure* moeten noemen; in Maple wordt het woord (type) 'function' voor functie*aanroep* gebruikt. Zie Module 26.

[9] Er is een verschil tussen de `->`-definitie en de definitie met `unapply`. Dit is er de oorzaak van dat bijvoorbeeld `f := x -> F;` hier niet goed zou werken. Zie §7.3.

Wanneer `->`, wanneer `unapply`? De `unapply`-variant is in twee gevallen handig:

- Om het resultaat van een (ingewikkelde) berekening verder als functie te kunnen gebruiken. Als men de `%`-operator wil gebruiken om een functie te definiëren, dan kan dat *alleen* met `unapply`.
- Als `F` een expressie is met een aantal verschillende parameters, en men wil verschillende functies hebben van één of meer van deze parameters afzonderlijk als variabele.

In alle andere gevallen is normaliter de 'pijltjesmethode' het handigst.

Voorbeeldsessie

Definitie van een functie:

```
> f := x -> x^2 + 1;
```

$$f := x \mapsto x^2 + 1$$

Functiewaarde:

```
> f(2);
```

$$5$$

F is een expressie:

```
> F := x^2 + a;
```

$$F := x^2 + a$$

```
> F(x); # geeft niet de functiewaarde:
```

$$(x(x))^2 + a(x)$$

Maak van F een functie van x of van a:

```
> f1 := unapply(F,x); f2 := unapply(F,a);
```

$$f1 := x \mapsto x^2 + a$$

$$f2 := a \mapsto x^2 + a$$

```
> f1(2);
```

$$4 + a$$

```
> f2(2);
```

$$x^2 + 2$$

Een functie van twee variabelen:

```
> v := (x,y) -> x^2 + y^3;
```

$$v := (x, y) \mapsto x^2 + y^3$$

```
> v(2,a);
```

$$4 + a^3$$

Een veel gemaakte fout:

```
> restart; f(x) := x^2 + 1;
```

$$f(x) := x^2 + 1$$

Dat ziet er goed uit, maar:

```
> f(2); f(y);
```

$$f(2)$$

$$f(y)$$

```
> f(x);
```

$$x^2 + 1$$

Toelichting

Let op het merkwaardige resultaat $\mathrm{x}(x)^2 + a(x)$ (maar geen foutmelding) van de (onzinnige) opdracht `F(x)`. Een verklaring hiervoor is te vinden in §6.9.
Let ook op het gebruik van de ronde haakjes bij de definitie van een functie van meer variabelen.
De voorbeeldsessie eindigt met een zeer veel gemaakte beginnersfout. De toekenning `f(x) := x^2 + 1` maakt van `f` géén functie; je kunt $f(2)$ of $f(y)$ niet uitrekenen. Er wordt alleen een 'Maple-object', dat wil zeggen: een variabele gemaakt met de naam '`f(x)`', en de waarde '`x^2 + 1`', en niets anders. Waarom Maple hier geen foutmelding geeft wordt uitgelegd in Module 30. ◇

> **!** Functies kunnen *niet* worden gedefinieerd door de toekenning `f(x) :=` *formule*.
> Men moet `f := x ->` *formule* gebruiken.

@ **Samenstelling van functies.** Maple kent ook de *samenstelling* van functies. Hiervoor dient het `@`-symbool. Dus $f \circ g(x)$ is in Maple `(f@g)(x)` (denk aan de haakjes). Ook de n-voudige samenstelling $f \circ f \circ \cdots \circ f(x)$ (n maal f toegepast op x) kent Maple: dit is `(f@@n)(x)`.

Voorbeeldsessie

Definitie van een functie:

```
> f := x -> x^2 + 1;
```

$$f := x \rightarrow x^2 + 1$$

Een tweede functie:

```
> g := x -> x+3;
```

$$g := x \rightarrow x + 3$$

Samenstelling:

```
> f@g;
```

$$f@g$$

```
> (f@g)(x);
```

$$(x+3)^2+1$$

Dit is hetzelfde als:

```
> f(g(x));
```

$$(x+3)^2+1$$

Nu f vier maal toegepast op a:

```
> (f@@4)(a);
```

$$(((a^2+1)^2+1)^2+1)^2+1$$

Toelichting

Op het eerste gezicht lijkt Maple de samenstelling `f@g` niet te ‘doen’. Dat lijkt maar zo, want op `(f@g)(x)` reageert het als verwacht. Er bestaat dus wel degelijk een functie $f \circ g$, alleen laat Maple hem nog niet zien als we hem zonder argument opvragen. $\diamond$

3.3 Constante functies

Aparte vermelding verdienen de *constante functies*.

Voorbeeldsessie

Een constante functie

```
> f := x -> 6;
```

$$f := x \to 6$$

```
> f(x);
```

$$6$$

```
> g := 6:  g(x);
```

$$6$$

```
> 6(x,y);
```

$$6$$

```
> h := x -> c;
```

$$h := x \to c$$

```
> h(x);
```

$$c$$

```
> j := c:
```

Hiermee wordt c nog geen functie!

```
> j(x); c(x);
```

$$c(x)$$

$$c(x)$$

Toelichting

We hebben de constante functie $x \mapsto 6$ op twee manieren gedefinieerd, namelijk met `f:= x -> 6` en met `g := 6`. Blijkbaar kan het getal `6` ook dienen als de *naam* van een *functie*. De aanroep `6(x)` heeft hetzelfde effect als `f(x)`. Hiermee wordt tevens verklaard waarom de typefout `3(4+5)` inplaats van `3*(4+5)` (Opgave 1.1) geen foutmelding oplevert.
Bij een constante functie $h : x \mapsto c$, met c een onbekende constante, ligt het iets anders. Nu werkt de toekenning `h := x->c` wél als verwacht en levert de aanroep `h(x)` iets anders dan `c(x)`. Dat komt omdat Maple niet kan 'weten' dat met `c` een constante wordt bedoeld. Meer hierover vindt u in Module 7. ◇

3.4 Stuksgewijs gedefinieerde functies

piecewise

Functies met verschillende functievoorschriften op verschillende deelintervallen kunnen met `piecewise` worden gemaakt. Hoe dat gaat kunnen we het beste demonstreren met een voorbeeld.

Voorbeeldopgave

De functie $f : \mathbb{R} \to \mathbb{R}$ wordt gedefinieerd door

$$f(x) = \begin{cases} 0 & \text{als } x < -1 \text{ of } x > 1 \\ x^2 & \text{als } -1 \le x \le 0 \\ x^3 & \text{als } 0 < x \le 1\,. \end{cases}$$

Maak een Maplefunctie volgens dit voorschrift.

Voorbeeldsessie

```
> f := x -> piecewise( x<-1,0, x<=0,x^2, x<=1,x^3, 0 );
```

$$f := x \mapsto \text{piecewise}(x < -1, 0, x \le 0, x^2, x \le 1, x^3, 0)$$

```
> f(t);
```

$$\begin{cases} 0 & t < -1 \\ t^2 & t \leq 0 \\ t^3 & t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

```
>  f(1/2);
```

$$1/8$$

Toelichting

Bij het gebruik van `piecewise` is het het handigst om de gehele reële as van links naar rechts 'af te werken'. Tussen de haakjes van `piecewise( )` komt steeds de voorwaarde in de vorm van een ongelijkheid, gevolgd door het bijbehorende voorschrift. Alleen het laatste voorschrift heeft geen voorwaarde nodig, wat dat wordt automatisch 'overige gevallen'.

`<=`, `>=` We zien hier tevens dat we voor het symbool "$\leq$" in Maple `<=` kunnen gebruiken. ◇

3.5 Decimale benaderingen, afronding

Een *decimale benadering* voor $f(a)$ kan worden verkregen met de
`evalf` opdracht `evalf(f(a))`. Iets preciezer: als `w` een expressie is, zorgt `evalf(w)` ervoor dat alle constanten die in `w` voorkomen decimaal benaderd worden. Als `w` zelf een constante is, dan krijgen we een decimale benadering van `w`.

Het aantal cijfers in deze benaderingen wordt bepaald door de varia-
`Digits` bele `Digits` (let op de hoofdletter!). Let op dat dit het *totaal* aantal cijfers aangeeft, dus het aantal cijfers voor de komma plus het aantal cijfers achter de komma. Deze heeft aan het begin van elke sessie de waarde 10, maar dit kan door de gebruiker worden veranderd. Een andere mogelijkheid is om het gewenste aantal cijfers als tweede argument van `evalf` op te geven. Zo levert bijvoorbeeld `evalf(f(a),15)` een benadering van $f(a)$ in 15 cijfers.

Het is ook mogelijk om Maple een ander (kleiner) aantal cijfers in de uitvoer op het scherm te laten tonen dan waarmee het de berekeningen uitvoert. Via Tools → Options... → Precision kunt u het getoonde aantal cijfers achter de komma instellen.

Er is bovendien nog een aantal Maple-functies beschikbaar om allerlei afrondingen naar gehele getallen te maken:

`round` `round(a)` berekent het gehele getal dat het dichtst bij a ligt;

`trunc, frac`	`trunc(a)`, `frac(a)` berekenen respectievelijk het gehele en het decimale deel van a als a een 'kommagetal' is. Als a een breuk is, bijvoorbeeld 4/3, dan heeft `trunc(a)` de waarde 1 en `frac(a)` de waarde 1/3.
`floor`	`floor(a)` berekent het dichtstbijzijnde gehele getal $\leq a$ (dit wordt ook wel de *entierfunctie* genoemd);
`ceil`	`ceil(a)` berekent het dichtstbijzijnde gehele getal $\geq a$.

Voorbeeldsessie

```
> evalf( sin(2) );
```

$$0.9092974268$$

```
> evalf( sin(2), 2 );
```

$$0.91$$

```
> Digits;
```

$$10$$

```
> Digits := 20;
```

$$20$$

```
> evalf( sin(2) );
```

$$0.90929742682568169540$$

```
> evalf( sin(a) );
```

$$\sin(a)$$

```
> Digits := 10:  evalf(2^1.1, 23);
```

$$2.143546925$$

Toch maar 10 cijfers!

```
> evalf(2^(11/10), 23 );
```

$$2.1435469250725863284260$$

Afrondingen:

```
> x := -2.51:
> round(x);
```

$$-3$$

```
> trunc(x);
```

$$-2$$

```
> frac(x);
```

$$-0.51$$

```
> ceil(x);
```

$$-2$$

```
> floor(x);
```

$$-3$$

Toelichting

Merk op dat `evalf(f(a))` niets doet, zolang `a` niet een vast getal is.

Als we het getal $2^{1.1}$ in 23 cijfers nauwkeurig willen uitrekenen dan gaat dat niet met `evalf(2^1.1, 23)` als `Digits` op 10 staat. De reden hiervan is dat dat Maple, zodra het de expressie `2^1.1` tegenkomt, er *meteen* 2.143546925 van maakt; vergelijk de voorbeeldsessie in §1.3. Dit (afgeronde) getal zou dan vervolgens in 23 cijfers nauwkeurig moeten worden weergegeven. Dit zou betekenen dat er hooguit een stel nullen achter komen; dit geeft geen extra informatie, dus laat Maple ze maar weg. ◇

Het 'omgekeerde' van `evalf` is ook mogelijk. Met

`convert, rational`

`convert( a, rational )`

wordt van het decimale getal `a`, dat door Maple dus als een *afgerond* getal wordt gezien, een breuk gemaakt, waarmee exact verder kan worden gerekend.

Voorbeeldsessie

```
> a := 3.14159:
> b := convert(a, rational);
```

$$b := \frac{76149}{24239}$$

```
> evalf(b);
```

$$3.141590000$$

Toelichting

Merk op dat `evalf(b)` weer een afronding van `b` op *Digits* (= 10) cijfers nauwkeurig maakt. ◇

3.6 Standaardfuncties, constanten

Een lijstje van standaardfuncties en hun Maple-equivalenten is opgenomen in tabel 3.1.
Andere wortels dan de vierkantswortel $\sqrt{x}$ verkrijgt men door de ^-operator te gebruiken. Zo is $\sqrt[3]{x}$ in Maple: `x^(1/3)`. Let daarbij op de haakjes(!), want `x^1/3` zou worden:

$$\frac{x^1}{3} = \frac{1}{3}\,x\,.$$

Zie verder §3.9.

`convert`

Met het commando `convert` kunnen we veel functies in een andere vorm krijgen.

functies	Maple-invoer
$x \mapsto \|x\|$	`x -> abs(x)`
$x \mapsto \sin x$	`x -> sin(x)`
$x \mapsto \sinh x$	`x -> sinh(x)`
$x \mapsto \cos x$	`x -> cos(x)`
$x \mapsto \cosh x$	`x -> cosh(x)`
$x \mapsto \tan x$	`x -> tan(x)`
$x \mapsto \arcsin x$	`x -> arcsin(x)`
$x \mapsto \arccos x$	`x -> arccos(x)`
$x \mapsto \arctan x$	`x -> arctan(x)`
$x \mapsto e^x$	`x -> exp(x)`
$x \mapsto \ln x$	`x -> ln(x)`
$x \mapsto \log x$	`x -> log(x)`
$x \mapsto \log_{10} x$	`x -> log[10](x)`
$x \mapsto \sqrt{x}$	`x -> sqrt(x)`
$x \mapsto x^a$	`x -> x^a` of `x -> x**a`

TABEL 3.1. Standaardfuncties en hun Maple-equivalenten. Let op: de functies `ln` en `log` zijn hetzelfde.

Voorbeeldsessie

```
> convert( sinh(x), exp );
```

$$\frac{1}{2}\,\mathrm{e}^{x} - \frac{1}{2}\,\mathrm{e}^{-x}$$

```
> convert( tan(x), sincos );
```

$$\frac{\sin(x)}{\cos(x)}$$

Toelichting

Inderdaad is

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

Raadpleeg `?convert` voor meer mogelijkheden. ◇

De functie arctan. De waarden van $\arctan x$ liggen (voor reële x) tussen $-\frac{\pi}{2}$ en $+\frac{\pi}{2}$. Naast deze 'normale' arctan kun je in Maple de functie `arctan` ook met *twee* argumenten aanroepen. Er geldt dan dat $-\pi < \arctan(y, x) \leq \pi$, en voor $x > 0$ is $\arctan(y, x) = \arctan(\frac{y}{x})$.

`arctan`

Er geldt dan bijvoorbeeld:

$\arctan(-1,-1) = -\frac{3}{4}\pi$	$\arctan(-1,0) = -\frac{1}{2}\pi$
$\arctan(-1,1) = -\frac{1}{4}\pi = \arctan(-1)$	$\arctan(0,1) = 0 = \arctan(0)$
$\arctan(1,1) = \frac{1}{4}\pi = \arctan(1)$	$\arctan(1,0) = \frac{1}{2}\pi$
$\arctan(1,-1) = \frac{3}{4}\pi$	$\arctan(0,-1) = \pi$

Als het tweede argument ≤ 0 is, dan is $\arctan(y,x)$ dus $\geq \frac{1}{2}\pi$ of $\leq -\frac{1}{2}\pi$.

`signum`

De functie signum. Behalve de bovengenoemde standaardfuncties noemen we nog de functie `signum`. Deze functie is gedefinieerd voor $x \neq 0$ als

$$\operatorname{signum}(x) = \frac{x}{|x|}.$$

Voor een reëel getal x levert de aanroep `signum(x)` dus 1 af als $x > 0$ en -1 als $x < 0$.

Pas op! verwar de functie `signum` niet met de functie `sign`, want dat is iets heel anders.

`Pi, I`
`exp(1)`
`infinity`

Constanten. De constanten π en i zijn in Maple respectievelijk `Pi` en `I` (let op de hoofdletters!). Voor het getal e moet men e^1 nemen, dus het Maple-commando `exp(1)`. In Maple wordt ∞ gerepresenteerd door de constante `infinity`.

`ininame`
`inifcn`

Een volledige lijst van de constanten en functies die Maple kent, kan worden verkregen door de helpopdracht te gebruiken: `?ininame` geeft een lijst van de constanten, en `?inifcn` van de functies.

Voorbeeldsessie

```
> f := x -> sin(x);
```

$$f := x \mapsto \sin(x)$$

```
> f(2);
```

$$\sin(2)$$

```
> evalf(f(2));
```

$$0.9092974268$$

```
> Digits := 20;
```

$$20$$

```
> evalf(f(2));
```

$$0.90929742682568169540$$

Let op: de basis van de natuurlijke logaritme is niet **e**:

```
> evalf(e);
```

$$e$$

... de letter e is een willekeurige variabele. Dit is te zien doordat Maple hem *cursief* afdrukt.

Inplaats daarvan gebruiken we:

```
> exp(1); evalf(%,6);
```

$$e$$

$$2.71828$$

Merk op dat “**exp(1)**” als “**e**” wordt afgedrukt.

Let ook op het onderscheid tussen **Pi** en **pi**:

```
> pi; evalf(%,6);
```

$$\pi$$

$$\pi$$

```
> Pi; evalf(%,6);
```

$$\pi$$

$$3.14159$$

3.7 Grafieken

plot

Om een grafiek te tekenen van een functie van één variabele dient het `plot`-commando. In Module 9 gaan we daar uitgebreid op in. Hier geven we een enkel eenvoudig voorbeeld.

We tekenen de grafiek van de functie $f(x) = 1 - x^2$ op het interval $[-1, 2]$. Dat kan op de volgende manieren:

```
plot( 1-x^2, x=-1..2 );
```

om een grafiek van de expressie “ $1-x^2$ ” te tekenen. In deze expressie komt de variabele x voor, die van -1 tot 2 moet lopen. Dit geven we aan met het tweede argument: `x=-1..2` in de plotopdracht.
Als we met `f := x -> 1-x^2` een *functie* hebben gemaakt, dan kunnen we de grafiek maken met

```
plot( f, -1..2 );
```

of, uiteraard

```
plot( f(t), t=-1..2 );
```

Merk hierbij op dat we de naam van de variabele niet op mogen geven als we in de plotopdracht alleen de naam van de functie noemen.
Als u de cursor *in* het plaatje brengt, dan kunt u met de rechter muisknop allerlei zaken veranderen.

3.8 Complexe getallen

evalc
Re, Im
conjugate
argument
csgn

Wanneer z een *complex getal* is, kan z in de vorm $x + iy$ geschreven worden met de opdracht `evalc`. Voor het reële deel en het imaginaire deel zijn de functies `Re` en `Im` beschikbaar. De geconjugeerde $\overline{z}$ is `conjugate(z)`. Voor de *hoofd*waarde van het argument kan `argument` (dus voluit geschreven!) worden gebruikt.
Verder noemen we nog de functie `csgn` (complexe signum-functie). Deze kan worden gebruikt om te bepalen of een complex getal in het rechter- of linkerhalfvlak van het complexe vlak ligt. Zie `?csgn` voor een precieze definitie.

Voorbeeldsessie

Een complexe functie:

```
> g := x -> exp(I*x);
```

$$g := x \mapsto e^{Ix}$$

```
> g(2);
```

$$e^{2\,I}$$

```
> evalc(g(2));
```

$$\cos(2) + I\sin(2)$$

```
> evalf(g(2));
```

$$-0.4161468365 + 0.9092974268\,I$$

Complexe getallen in Cartesische vorm:

```
> z := 1 + 2*I: Re(z); Im(z);
```

$$1$$
$$2$$

```
> abs(z); argument(z);
```

$$\sqrt{5}$$
$$\arctan(2)$$

```
> Re(a+b*I);
```

$$Re\,(a + Ib)$$

Polaire vorm:

```
> z1 := polar(2,Pi/4); evalc(z1);
```

$$z1 := \mathrm{polar}(2, 1/4\,\pi)$$
$$\sqrt{2} + I\sqrt{2}$$

```
> z2 := 2 * exp( Pi/4*I); convert(z2, polar);
```

$$z2 := \sqrt{2} + I\sqrt{2}$$
$$\mathrm{polar}(2, 1/4\,\pi)$$

```
>  evalc(z1-z2);
```

$$0$$

Toelichting

Vele van de functies genoemd in tabel 3.1 alsmede de functie `signum` werken ook op complexe getallen. Merk bovendien op dat `evalf` ook werkt om van e^{2i} een numerieke benadering te krijgen.
Het lijkt erop dat Maple het reële deel en het imaginaire deel van $a + b\,i$ niet wil geven. Echter, zolang niet bekend is dat a en b zelf *reële* getallen zijn, kan Maple niet meer vertellen dan het nu doet.
Een complex getal in polaire vorm kan naar believen worden opgegeven met de `exp`-functie of met de functie `polar`.

`polar`

Als z een complex getal is, dan (her)schrijft `evalc(z)` het in Cartesische vorm en `polar(z)` het in polaire vorm. ◇

3.9 Complexe wortels en logaritme

Het is gebruikelijk om $\sqrt{a}$ alleen te definiëren voor positieve (reële) waarden van a. Er geldt dan dat met $\sqrt{a}$ de *positieve (reële)* oplossing van $x^2 = a$ wordt bedoeld.
Voor de functie $x \mapsto \sqrt[n]{x}$, met $n \in \mathbb{N}\backslash\{0\}$ zijn er minstens twee mogelijkheden voor het domein:

(1) $x \in \mathbb{R}, x \geq 0$ voor alle n; óf:
(2) $x \in \mathbb{R}, x \geq 0$ als n *even* is, en $x \in \mathbb{R}$ als n *oneven* is.

In beide gevallen is $\sqrt[n]{a}$ uiteraard een oplossing van de vergelijking $x^n = a$.
Als we de wortelfunctie willen uitbreiden naar de complexe getallen wordt het ingewikkelder. De vergelijking $x^n = a$ met $a \neq 0$ heeft in $\mathbb{C}$ n verschillende oplossingen, en er zijn dus evenzoveel mogelijkheden om $\sqrt[n]{a}$ te definiëren. In de praktijk worden drie definities gebruikt:

(1) de (complexe) oplossing van $x^n = a$ met de *kleinste absolute waarde* van het argument (positief argument als er twee mogelijkheden zijn). In dit geval is
$$\sqrt[3]{-1} = \frac{1}{2} + \frac{1}{2}\,i\,\sqrt{3}.$$
(2) de (complexe) oplossing van $x^n = a$ waarvan het argument zo weinig mogelijk verschilt van het argument van a. In dit geval is
$$\sqrt[3]{-1} = -1.$$

(3) de verzameling van *alle* oplossingen van de vergelijking $x^n = a$. In dit geval is

$$\sqrt[3]{-1} = \left\{ \frac{1}{2} - \frac{1}{2}\,i\,\sqrt{3},\ -1,\ \frac{1}{2} + \frac{1}{2}\,i\,\sqrt{3} \right\}.$$

Met definitie 3 is $\sqrt[n]{x}$ uiteraard geen functie volgens de in §3.1 gegeven definitie. We spreken van een *meerwaardige functie*.

Een vergelijkbare situatie bestaat als we $\ln a$ voor $a \in \mathbb{C}$ definiëren als oplossing van $e^z = a$. Met $a = r\,e^{i\varphi}$ is dan $\ln a = \ln r + i\varphi + 2k\pi i$, met $k \in \mathbb{Z}$. Ook dit leidt tot een definitie van $\ln a$ als meerwaardige functie. Het is gebruikelijk om voor $\ln a$ de *hoofd*waarde te nemen: $\ln|a| + i \arg a$ waarbij $-\pi < \arg a \leq \pi$.
Gebruikmakend van de aldus gedefinieerde logaritme kan men z^p voor $z \in \mathbb{C}$ definiëren als $z^p = e^{p \ln z}$. Met $p = 1/n$, dus $z^p = \sqrt[n]{z}$ krijgt men hiermee precies definitie 1 voor de wortel (als tenminste de hoofdwaarde van de ln wordt genomen).

3.10 Wortels en logaritme

In Maple zijn alle standaardfuncties in principe gedefinieerd voor complexe argumenten. Er zijn daarom twee wortelfuncties beschikbaar: `root`, de wortel volgens definitie 1 en `surd`, volgens definitie 2. De expressie `a^(1/n)` voor $\sqrt[n]{a}$ wordt geïnterpreteerd als `root(a,n)`, ook in te geven als `root[n](a)`.

`root`
`surd`

De expressie `RootOf(z^n - a)` kan gebruikt worden als een niet nader gespecificeerde oplossing van de vergelijking $z^n = a$ wordt bedoeld en lijkt daarom het meest op definitie 3. Zie hiervoor de voorbeeldsessie en Module 5.

`RootOf`

Voorbeeldsessie

De 'standaard'-derdemachtswortel van Maple:

```
> (-1)^(1/3):
```

is hetzelfde als

```
> root(-1,3);
```

$$\sqrt[3]{-1}$$

```
> evalc(%);
```

$$1/2 + 1/2\,\mathrm{I}\sqrt{3}$$

Een (zo mogelijk) reële derdemachtswortel:

```
> surd(-1,3);
```

$$-1$$

Een oplossing van de vergelijking $z^3 + 1 = 0$:

```
> a := RootOf( z^3 + 1 );
```

$$a := RootOf\left(\_Z^3 + 1\right)$$

met a kan gewoon worden gerekend:

```
> simplify( 2*a^3 + 4 );
```

$$2$$

Alle oplossingen van de vergelijking $z^3 + 1 = 0$:

```
> allvalues(a);
```

$$1/2 + 1/2\,\mathrm{I}\sqrt{3},\ -1,\ 1/2 - 1/2\,\mathrm{I}\sqrt{3}$$

Een oplossing van de vergelijking $e^z = -1$:

```
> ln(-1);
```

$$\mathrm{I}\pi$$

Toelichting

Men moet erop verdacht zijn dat `a^(1/n)` bij negatieve waarden van a een niet-reëel antwoord geeft, óók als n oneven is. Als u er zeker van wilt zijn dat $\sqrt[n]{a}$ reëel is als a reëel is, kunt u beter `surd` gebruiken. Zie ook opgave 3.10.

`allvalues` De functie `allvalues` kan worden gebruikt om een uitdrukking waarin `RootOf` voorkomt verder uit te werken. Zie verder §5.5. ◇

`RealDomain` Overigens is het ook mogelijk Maple te forceren om alléén reëel te rekenen. Men doet dat door de *bibliotheek* `RealDomain` te laden.

Voorbeeldsessie

Laden van de bibliotheek

```
> with(RealDomain);

[Im, Re, ^, arccos, arccosh, arccot, arccoth, arccsc, arccsch,
arcsec, arcsech, arcsin, arcsinh, arctan, arctanh, cos, cosh,
cot, coth, csc, csch, eval, exp, expand, limit, ln, log, sec,
sech, signum, simplify, sin, sinh, solve, sqrt, surd, tan,
tanh]

> a := (-1)^(1/3);
```

$$a := -1$$

```
> ln(-1);
```

$$\mathit{undefined}$$

```
> unwith(RealDomain);
> (-1)^(1/3); evalc(%);
```

$$\sqrt[3]{-1}$$

$$1/2 + 1/2\,\mathrm{I}\sqrt{3}$$

Alternatief

```
> use RealDomain in (-1)^(1/3); ln(-1) end use;
```

$$-1$$

$$\mathit{undefined}$$

Toelichting

`with` Met het commando `with( )` wordt een bibliotheek geladen. Als het commando met een puntkomma wordt afgesloten, reageert Maple met een lijst van functies die vanaf dit moment beschikbaar zijn. In dit geval zijn het allemaal functies die een andere werking hebben gekregen. Zo is bijvoorbeeld $\ln x$ niet gedefinieerd (*undefined*) als Maple er achter kan komen dat x negatief is. De functies krijgen weer `unwith` hun oorspronkelijke werking met de opdracht `unwith(RealDomain)` waardoor deze bibliotheek weer wordt verwijderd.
Als men alleen maar voor een paar opdrachten reëel wil rekenen, dan `use` kan dat met `use`. ◇

3.11 Opmerking

`->` Een uitdrukking die is gedefinieerd via het pijltje `->` wordt door Maple als een procedure opgevat. Dus in de expressie `f(x)` is `f` de naam van de procedure, en `x` het argument. Het resultaat van de aanroep `f(x)` is een uitdrukking in `x`.
En we herhalen de waarschuwing van §3.2 nog maar eens:

!	In het algemeen is het dus niet goed om een functie in te voeren als bijvoorbeeld `f(x) := x^2`. Hoe dit precies zit komt aan de orde in Module 30. Men moet óf `f` als een *expressie* definiëren via `f := x^2` óf als een functie via `f := x -> x^2`.

Opgave 3.1

Bereken de 22^e decimaal van $\ln(7)$, e^5, $2^{7.1}$, π.

Opgave 3.2

Gebruik `?inifcn`; bereken daarna 25! en de binomiaalcoëfficiënt $\binom{32}{19}$.

Opgave 3.3

Bereken achtereenvolgens 5!, $\Gamma(6)$, $\Gamma(6.5)$, $\Gamma(7)$, 6!
(gebruik `?inifcn`; Γ is de Griekse hoofdletter gamma). Wat voor rol zou $\Gamma(x)$ voor $x \geq 1$ kunnen spelen?

Opgave 3.4

Gebruik het commando `simplify` om $\sqrt{x^2}$ te vereenvoudigen. Ga na of het antwoord klopt voor $x > 0$ en voor $x < 0$.
Doe hetzelfde nadat u de bibliotheek `RealDomain` hebt geladen.
Wat is het resultaat van `sqrt(x^2, symbolic)`? (met en zonder `RealDomain`). In welke gevallen zou u deze optie kunnen gebruiken?

Opgave 3.5

(a) Bereken $y - \sin(\arcsin(y))$. Ga na dat het antwoord klopt voor alle mogelijke waarden van y;
(b) Bereken nu $y - \arcsin(\sin(y))$. Had u dit antwoord verwacht? Vereenvoudig het antwoord eventueel met `simplify(%)`; heeft het nut om de bibliotheek `RealDomain` te laden?
(c) Vereenvoudig de uitdrukking $y - \arcsin(\sin(y))$ ook eens met `simplify(%,symbolic)`, zie Module 4.
(d) Definieer $y := 3\pi/4$, en bereken opnieuw $y - \arcsin(\sin(y))$. (Waarom) klopt het antwoord?

Opgave 3.6

(a) Bepaal

$$\mathrm{Re}\left(\frac{2+i}{1-i}\right) \quad \text{en} \quad \mathrm{Im}\left(\frac{2+i}{1-i}\right)$$

(b) Schrijf de volgende getallen in de vorm $x + iy$:

$$\frac{1-2i}{3+4i} - \frac{2+i}{5i}, \quad \frac{1}{(1-i)(1-2i)(1+2i)}$$

Opgave 3.7

Schrijf de volgende complexe getallen in polaire vorm:
$-3, \quad 2i, \quad -\sqrt{3} - i, \quad -1 - i\sqrt{3}.$

Opgave 3.8

Gegeven is het complexe getal $z = \sqrt{3}(1+i) + 1 - i$.

(a) Bereken z^{12};

(b) Bepaal $\arg(z)$.
Aanwijzing: Bepaal eerst $\arg(z^{12})$.

Opgave 3.9

Definieer, met zo weinig mogelijk tikwerk, de functie

$$h : x \mapsto \frac{(x^2 + 3\log(x))^5 + (x^2 + 3\log(x))^3 + 1}{(x^2 + 3\log(x))^4},$$

en bereken $h(1)$, $h(2)$ en $h(h(h(h(1))))$.
Aanwijzing: Definieer h als samenstelling van twee functies. Uw uitwerking moet dus van de vorm zijn:

```
> f := x -> ...
> g := y -> ...
> h := g@f;
```

Opgave 3.10

Onderzoek of Maple de expressie `a^(p/q)` interpreteert als $\sqrt[q]{a^p}$ dan wel als $(\sqrt[q]{a})^p$.
Aanwijzing: Neem $a = -1$, $p = 2$ en $q = 3$.