Module 4 Vereenvoudigen en substitutie

Onderwerp	Vereenvoudigen, combineren en uitschrijven van formules.
Voorkennis	VWO-stof over optellen en vermenigvuldigen van trigonometrische, exponentiële en logaritmische functies.
Expressies	`simplify, expand, combine, factor, symbolic, collect, normal, rationalize, assume, ::, subs, type, whattype, algsubs, eval, convert`
Zie ook	Module 11, Module 26

4.1  Schrijfwijzen voor wiskundige expressies

Vaak kunnen we wiskundige expressies op diverse manieren schrijven, bijvoorbeeld:

$x^2 - y^2$	of	$(x-y)(x+y)$
$\sin x \cos y + \sin y \cos x$	of	$\sin(x+y)$
$e^x e^y$	of	e^{x+y}
$(a^x)^y$	of	a^{xy}
$\log x + \log y$	of	$\log(xy)$
$\sqrt{4}+1$	of	3

De expressies aan de rechterkant van de tabel zijn zó geformuleerd dat het aantal 'ingewikkelde' functieaanroepen zo klein mogelijk is. Bijvoorbeeld, één keer vermenigvuldigen is makkelijker dan twee keer machtsverheffen, één sinus uitrekenen is minder werk dan twee sinussen en twee cosinussen enzovoort.

4.2  Herschrijfregels in Maple

`simplify`

Vereenvoudigen. De meest gebruikte opdracht om een formule te vereenvoudigen is `simplify`. Het gebruik hiervan is aan te raden om te proberen elke expressie die er nodeloos ingewikkeld uitziet te vereenvoudigen.

Voorbeeldsessie

```
> b := (x^3-y^3)/(x-y);
```

$$b := \frac{x^3-y^3}{x-y}$$

```
> simplify(b);
```

$$y^2 + xy + x^2$$

Pas op! De waarde van **b** is niet veranderd.

```
> b;
```

$$\frac{x^3-y^3}{x-y}$$

Als we **b** willen vervangen door de vereenvoudigde versie ervan moeten we deze waarde expliciet aan de variabele **b** toekennen. Bijvoorbeeld zo:

```
> simplify(b):  b := %;
```

$$y^2 + xy + x^2$$

Toelichting

Het commando `simplify(b);` geeft een vereenvoudiging van de expressie `b` als *output*; de variabele `b` zélf wordt door dit commando niet veranderd. ◇

Soms is het handig om aan `simplify` een *nevenvoorwaarde* op te geven, waardoor we uitdrukkingen waarin een bepaalde expressie vaak voorkomt eenvoudiger kunnen schrijven. Zo levert bijvoorbeeld

```
simplify( x*y*z + x*y^2, {x*y=p} );
```

het resultaat $pz + py$.

`trigsubs`

Bij expressies die goniometrische functies bevatten, zou men eens in een 'formulelijst' kunnen zoeken. Het commando `trigsubs(p)`, met `p` een goniometrische expressie, levert een lijstje van uitdrukkingen waaraan `p` gelijk is. Mocht u dus bijvoorbeeld de 'dubbele hoek formule' vergeten zijn: `trigsubs(sin(t)^2)` verschaft u hem. Vervolgens zou u deze kunnen gebruiken als nevenvoorwaarde om met `simplify` een uitdrukking met $\sin^2(t)$ erin te herschrijven.

`expand` `factor`

Haakjes uitwerken en factoriseren. Om haakjes uit te werken gebruikt men in het algemeen de opdracht `expand`. Een polynoom wordt in factoren ontbonden met de opdracht `factor`. Zie hiervoor verder Module 11.

Voorbeeldsessie

```
> p := (x-1)*(x-y)*(x+4);
```

$$p := (x-1)\,(x-y)\,(x+4)$$

```
> q := expand(p);
```

$$q := x^3 + 3\,x^2 - x^2y - 3\,xy - 4\,x + 4\,y$$

```
> factor(q);
```

$$(x-1)\,(x-y)\,(x+4)$$

```
> r := sin(x+y);
```

$$r := \sin(x+y)$$

```
> s := expand(r);
```

$$s := \sin(x)\cos(y) + \cos(x)\sin(y)$$

```
> factor(s);
```

$$\sin(x)\cos(y) + \cos(x)\sin(y)$$

```
> t := exp(x+y);
```

$$t := \mathrm{e}^{x+y}$$

```
> expand(t);
```

$$\mathrm{e}^{x}\mathrm{e}^{y}$$

Toelichting

In dit voorbeeld worden bij drie soorten formules 'de haakjes' uitgewerkt. Het commando `factor` doet alleen bij polynomen 'het omgekeerde' van `expand`. ◇

Andere procedures om polynomen te herschrijven zijn bijvoorbeeld `collect` en `normal` (zie Module 11).

`combine`

Combineren. Het commando `combine` heeft eventueel een 'hint' nodig om te weten wat voor soort functies gecombineerd moeten worden. Daarvoor dienen extra argumenten, bijvoorbeeld `exp`, `ln`, `power` en `trig`.

Voorbeeldsessie

```
> p := exp(x)*sin(x)*sin(y) * exp(2*x)*cos(x)*cos(y);
```

$$p := \mathrm{e}^{x}\sin(x)\sin(y)\,\mathrm{e}^{2\,x}\cos(x)\cos(y)$$

```
> combine(p);
```

$$1/8\,\mathrm{e}^{3\,x}\cos(2\,x-2\,y) - 1/8\,\mathrm{e}^{3\,x}\cos(2\,x+2\,y)$$

```
> combine(p,trig);
```

$$1/8\,\mathrm{e}^{x}\mathrm{e}^{2\,x}\cos(2\,x-2\,y) - 1/8\,\mathrm{e}^{x}\mathrm{e}^{2\,x}\cos(2\,x+2\,y)$$

```
> combine(p,exp);
```

$$\sin(x)\sin(y)\cos(x)\cos(y)\,\mathrm{e}^{3\,x}$$

```
> q := ln(x) - ln(2);
```

$$q := \ln(x) - \ln(2)$$

```
> combine(q);
```

$$\ln(1/2\,x)$$

```
> r := ln(x) - ln(y);
```

$$r := \ln(x) - \ln(y)$$

```
> combine(r,ln);
```

$$\ln(x) - \ln(y)$$

```
> combine(r) assuming positive;
```

$$\ln\left(\frac{x}{y}\right)$$

Toelichting

We zien dat we de opdracht `combine` alléén op de goniometrische functies sin en cos kunnen laten werken door het extra argument (optie) `trig` (van *trigonometric*) toe te voegen. Om de werking te beperken tot de e-machten dient de optie `exp`.
We zien tevens dat $\ln x - \ln y$ niet zonder meer gecombineerd wordt tot $\ln(\frac{x}{y})$. Dat komt omdat dit niet voor alle waarden van x en y klopt, bijvoorbeeld als x en y beide negatief zijn. We krijgen deze vereenvoudiging wel voor elkaar als we vermelden dat alle gebruikte variabelen positief verondersteld worden. Zie verder de volgende paragraaf. ◇

Andere opties voor de `combine`-opdracht zijn onder andere: `power` (voor machten), `radical` (voor wortels) en `arctan`.

Wortels in de noemer. Sommigen hebben een hekel aan breuken met wortels in de noemer. Deze kunnen vaak worden weggewerkt door teller en noemer met een geschikte factor te vermenigvuldigen.
`rationalize` Maple heeft daarvoor het commando `rationalize`.

Voorbeeldsessie

```
> p := 1/(sqrt(2)-1);
```

$$p := \frac{1}{-1+\sqrt{2}}$$

```
> rationalize(p);
```

$$1+\sqrt{2}$$

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

$$\frac{1}{2}\sqrt{2}$$

Toelichting

We zien dat Maple zelfs $\frac{1}{\sqrt{2}}$ direct verandert in $\frac{1}{2}\sqrt{2}$. ◇

4.3 Vereenvoudigen met veronderstellingen

Vaak zullen voor een mens zeer voor de hand liggende vereenvoudigingen door de machine niet worden uitgevoerd. Soms is daarvan de reden dat zo'n vereenvoudiging niet in alle gevallen geoorloofd is. Hierbij moeten we in het oog houden dat Maple er normaliter van uitgaat dat alle getallen complex zijn. Zo wordt bijvoorbeeld $\ln x$ geïnterpreteerd als "een getal $z \in \mathbb{C}$ met $e^z = x$". In dat geval mag x ook negatief zijn. En met bijvoorbeeld $x = y = -2$ geldt natuurlijk niet meer dat $\ln x + \ln y = \ln(xy)$ als men de *hoofd*waarde van de ln-functie neemt (zoals Maple doet, zie ook §3.9). In veel gevallen zullen we daarom bij vereenvoudigingen (en ook bij andere berekeningen) verschillende *aannames* over de gebruikte variabelen willen doen. Dat kan op diverse manieren met `assume` of `assuming`. We laten dat zien aan de hand van een paar voorbeelden.

`assume`

Voorbeeldsessie

```
> sqrt(a^2*x^2);
```

$$\sqrt{a^2x^2}$$

```
> simplify(%);
```

$$\sqrt{a^2x^2}$$

Twee manieren om te aan te geven dat a positief en x negatief is:

```
> sqrt(a^2*x^2) assuming a::positive, x::negative;
```

$$-ax$$

```
> sqrt(a^2*x^2) assuming a>0, x<0;
```

$$-ax$$

Beide variabelen reëel

```
> simplify( sqrt(a^2*x^2) ) assuming real;
```

$$|ax|$$

Alleen a reëel (en x complex):

```
> simplify( sqrt(a^2*x^2) ) assuming a::real;
```

$$\mathrm{csgn}\,(x)\,x\,|a|$$

Paardemiddel:

```
> simplify( sqrt(a^2*x^2), symbolic );
```

$$ax$$

Gehele veelvouden van π:

```
> simplify( sin(n*Pi) );
```

$$\sin\,(n\pi)$$

```
> sin(n*Pi) assuming n::integer;
```

$$0$$

```
> cos(n*Pi) assuming n::integer;
```

$$(-1)^n$$

```
> cos(n*Pi) assuming n::odd;
```

$$-1$$

```
> assume(n,integer);
> n^2 + 2*n + 1;
```

$$n\sim^2 + 2\,n\sim + 1$$

```
> sin(n*Pi);
```

$$0$$

Toelichting

In de `assuming`-vorm kan *per statement* worden aangegeven welke veronderstellingen over verschillende variabelen Maple moet gebruiken bij de berekening. Let op de 'dubbele' dubbele punt (`::`).

`::`

Door bij het commando `combine( )` of `simplify( )` de extra optie `symbolic` mee te geven schakelen we als het ware alle voorzichtigheid bij het vereenvoudigen uit. Maple maakt dan van $\sqrt{x^2}$ gewoon x, zonder er rekening mee te houden dat dit niet klopt als niet $x \geq 0$ is, en $\ln x + \ln y$ wordt zonder meer $\ln xy$. Als u deze optie gebruikt, dan zult u zélf altijd moeten nagaan of het resultaat geldt voor de alle waarden die de variabelen kunnen aannemen.

`symbolic`

Het is ook mogelijk om door middel van `assume( )` aan te geven wat hierna *steeds* verondersteld moet worden. Dit is vooral nuttig als in de formules fysische grootheden zoals bijvoorbeeld de tijd t (≥ 0), de massa m (idem) en dergelijke worden gebruikt. Raadpleeg `?assume` en `?assuming` om te weten te komen wat er allemaal mogelijk is.

Dat van bepaalde variabelen iets is verondersteld maakt Maple duidelijk door ze met een 'kringeltje' ($\sim$) erachter weer te geven. Dat kan met de Options-knop desgewenst worden uitgezet. ◇

Ten slotte een aanbeveling. Doe niet ál te veel moeite om Maple zover te krijgen om ingewikkelde formules te vereenvoudigen. Vaak kunt u het met potlood en papier zelf sneller. En als er met de formule verder gerekend moet worden heeft het vaak ook helemaal geen zin om te vereenvoudigen, want Maple rekent met ingewikkelde formules meestal even gemakkelijk als met eenvoudige formules.

4.4  Iets over typering

De uitdrukking `assuming x::negative` betekent: "veronderstel dat de variabele x van het *type* 'negatief (reëeel) getal' is."
In feite heeft élke expressie (dus niet alleen getallen) in Maple een type, waarmee aangegeven wordt wat voor soort expressie dit is. Het type van een expressie kan worden opgevraagd met het commando `whattype`. We geven een paar voorbeelden:

`whattype`

`whattype(3)`	`integer`
`whattype(3.0)`	`float`
`whattype(a)`	`symbol`
`whattype("a")`	`string`
`whattype( f(a) )`	`function`
`whattype(a+3)`	`+`
`whattype(a*b)`	`*`
`whattype(a^b)`	`^`
`whattype(a*b+3=x^2)`	`=`

De diverse typen zijn gerangschikt in een hiërachie, de zogenaamde *type-hiërarchie*. Bijvoorbeeld, een expressie van het type `integer` is automatisch ook van het type `algebraic`. We zeggen dat het type `integer` een *subtype* is van `algebraic`, en `algebraic` een *supertype* van `integer`.
Te midden van alle mogelijke typen die Maple kent is er een collectie zogenaamde *basistypen*. Tot deze categorie behoren onder andere de typen `integer`, `fraction`, `float` (dat is een getal met een decimale punt erin), `symbol`, `string`, `function`, `+`, `*` en `^`. Men kan met `?whattype[surface]` opvragen welke hier nog meer toe behoren. De overige typen die Maple kent (dit zijn dus sub- of supertypen van een of meer basistypen) komt men te weten via `?type`. Alle typen die niet tot de basistypen behoren zijn afgeleid van deze basistypen, hetzij door combinatie, hetzij door specialisatie[10].
Bijvoorbeeld, een expressie is van het type `rational` als zij van een van de basistypen `integer` of `fraction` is (combinatie). Een expressie van het type `Array` (zie Module 8) is automatisch van het basistype `rtable`, omdat een `Array` in feite hetzelfde is als een `rtable` die aan enkele extra eisen voldoet (specialisatie).
Basistypen kunnen dus zowel als subtype, als als supertype optreden. De procedure `whattype` geeft altijd een van de basistypen. Een

[10]Het is ook mogelijk zelf eigen typen te definiëren, zie Module 28.

expressie kan in het algemeen van verscheidene typen zijn, maar is slechts van één basistype. Om te testen of een expressie van een zeker sub- of supertype is (bijvoorbeeld om na te gaan of de expressie `3` van het type `numeric` is), dient de procedure `type`:

`type`

`a := 2:`	`type(a, integer)`	`true`
	`type(a, numeric)`	`true`
`a := p+q:`	`type(a, '+')`	`true`
`a := p=q:`	`type(a, equation)`	`true`

Let op dat bij het onderzoek of een expressie van het type `+`, `*` of `^` is, deze symbolen tussen *back-quotes* gezet moeten worden.
Bij het gebruik van `type` wordt van elk niveau waarop het eerste argument van `type` kan worden geëvalueerd, gekeken of dit van het type is dat is opgegeven als tweede argument van `type`. Dit verklaart de volgende sessie:

Voorbeeldsessie

```
> a := 2:  whattype(a);
```

integer

```
> type(a,constant); type(a,numeric);
```

true

true

```
> type(a,rational); type(a,algebraic);
```

true

true

```
> type(a,float); type(a,even);
```

false

true

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

$$f := x \to x^2$$

```
> whattype(f);
```

symbol

```
> whattype(eval(f));
```

procedure

```
> type(f,symbol);
```

true

```
> type(f,procedure);
```

true

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

'^'

g is nog niet gedefinieerd:

```
> whattype(g(x));
```

function

```
> whattype(Pi);
```

symbol

```
> type(Pi, numeric); type(Pi, constant);
```

false

true

Toelichting

De argumenten van `whattype` worden volgens de normale regels geëvalueerd. Een procedurenaam evalueert dus tot een naam (zie §25.2, blz. 392), en namen zijn (meestal) van het basistype `symbol`. Na toepassing van `eval` blijkt dat een volledige evaluatie van `f` in een procedure resulteert. Procedures zijn van het basistype `procedure`. De procedure `type` controleert voor elk van deze evaluaties van `f` of deze van het als tweede argument opgegeven type zijn (`symbol` respectievelijk `procedure`).

Maple-constanten zoals `Pi` zijn enigszins problematisch. Omdat de typen `numeric` en `constant` geen basistypen zijn, reageert Maple op `whattype(Pi);` slechts met `symbol`. Dat het getal(!) π door Maple niet als `numeric` wordt beschouwd zou men kunnen betreuren; het heeft te maken met de wijze waarop formules waarin π voorkomt worden vereenvoudigd. Als men bijvoorbeeld van een actuele parameter in een procedures wil testen of het een getal is (zie §29.4) moet men hiermee rekening houden: men zou in plaats van `type(a,numeric)` moeten vragen: `type(a,{constant,numeric})`.[11] ◇

4.5 Substitutie

`subs`

We kunnen in een expressie ook variabelen vervangen door andere. Hiervoor dient de opdracht `subs`. Bijvoorbeeld `subs( x=y, x+1 )` betekent: "vervang in de expressie `x+1` de naam `x` door de naam `y`" en levert dus `y+1`.
Wanneer we in een expressie meer dan één variabele willen vervangen, dan kan dat ook. Bijvoorbeeld

`subs( {x=y,y=z}, x+2*y+3*z )` levert $y + 5z$, en
`subs( {x=y,y=x}, x+2*y )` levert $y + 2x$.

[11] Of in de heading moeten opgeven: `a::{constant,numeric}`.

We zien dat dit gebruik van `subs` de bedoelde substituties simultaan uitvoert. Dit wordt bewerkstelligd door de accolades. Wanneer we de accolades weglaten, worden de substituties na elkaar uitgevoerd, de meest linkse het eerst. Dus

`subs( x=y, y=x, x+2*y )` levert $3x$, en
`subs( y=x, x=y, x+2*y )` levert $3y$.

Het kan voorkomen dat `subs` niet werkt zoals men zou verwachten. Bijvoorbeeld `subs( x*y=z, x*y*z )` levert niet z^2. Dit heeft te maken met de manier waarop Maple intern haar informatie opslaat. We zullen dit hier niet verder bespreken. De belangstellende lezer wordt verwezen naar Module 26.

`algsubs` Overigens zou in dit voorbeeld het commando `algsubs` uitkomst brengen. Zie `?algsubs` voor details.

> **!** Men moet erop verdacht zijn dat in `subs( x=y, x+2*y )` de `x` *niet* als '*dummy*' beschouwd mag worden. Zo zal na `x := 2:` het commando `subs( x=y, x+2*y )` geïnterpreteerd worden als `subs( 2=y, 2+2*y )` en resulteren in het antwoord $y + y^2$ (de `2` vervangen door `y`).

`eval` Het `subs`-commando doet niets anders dan in een expressie het ene 'teken' te vervangen door een ander. Het commando `eval` lijkt daar veel op, maar kan beter worden geïnterpreteerd als: "bereken de expressie waarin x voorkomt voor een gegeven waarde van x." We geven een voorbeeld.
Het commando

```
subs( x=Pi/2, sin(x) )
```

levert

$$\sin(\frac{\pi}{2}),$$

terwijl

```
eval( sin(x), x=Pi/2 )
```

direct het antwoord 1 geeft.
Men zou kunnen zeggen dat `eval` over enige 'wiskundige intelligentie' beschikt, terwijl `subs` alleen maar 'dom' tekens kan vervangen. Spectaculairder voorbeelden volgen nog (zie §6.9).

Opgave 4.1

Vereenvoudig de uitdrukking

$$\sin(2x+1)\cos(3x-5) + \sin(3x-5)\cos(2x+1).$$

Opgave 4.2

Bepaal het definitiegebied van de functie

$$\log((x^3 - 36x^2 + 431x - 1716)^2) + \log((x^2 - 29x + 210)^2)$$

Aanwijzing: Onderzoek waar de expressies onder het log-symbool nul zijn.

Opgave 4.3

Vereenvoudig $\cos(2\arctan(t))$.

Opgave 4.4

Raadpleeg `?convert` om

$$\frac{\tan(x)}{1 + 2\tan^3(x)}$$

uit te drukken in $\sin(x)$ en $\cos(x)$.

Opgave 4.5

Transformeer met `subs` de expressie $x + y + 3z$ naar $a + y + 3b$.

Opgave 4.6

Gegeven de expressie $p = ax^4 + bx^2 + c$.
Substitueer $x^2 = w$ in p. Gebruik desnoods `algsubs`, maar met `subs` gaat het óók (hoe?).