

Expressies en Procedures

Version 6.0.1

September 4, 2014

1 Prefixnotatie

1.1 Van infix- naar prefixnotatie

Schrijf op een blad papier de volgende expressies in Scheme-notatie (prefixnotatie). Indenteer om de leesbaarheid te verhogen.

- $\frac{a+b}{e} - \frac{c+d}{f}$
- $c + \frac{a}{b \times c + \frac{d}{e + \frac{f}{g}}}$
- $\frac{a + \frac{b}{c}}{d} \times \frac{e}{\frac{g}{i} - h}$

1.2 Verifieer syntax

Verifieer je antwoorden uit deel §1.1 “Van infix- naar prefixnotatie” door ze uit te proberen in een Scheme vertolker.

1.3 Maak Nul

Bedenk waarden voor de parameters a tot en met h zodat de eerste twee expressies 0 en de laatste 1 voorstellen.

1.4 Verifieer waarden

Verifieer opnieuw de correctheid van je antwoord uit deel §1.3 “Maak Nul” door het uit te proberen in Scheme.

2 Syntax en evaluatie van expressies

2.1 Voorspel het resultaat van enkele eenvoudige expressies

Voorspel (zonder gebruik te maken van een Scheme vertolker) voor elk van onderstaande expressies het resultaat indien ze in de gegeven (alfabetische) volgorde geëvalueerd worden. Leg de gevallen uit waarbij het misloopt.

Controleer nadien, door middel van een Scheme vertolker, of je prognoses correct was. Probeer de fouten die je gemaakt hebt te begrijpen en uit te leggen.

1. `(* (+ 2 2) 5)`
2. `(* (+ 2 2) (5))`
3. `(* (+ (2 2) 5))`
4. `(* (+ 22) 5)`
5. `(5 * 4)`
6. `5*4`
7. `(define 5*4 20)`
8. `5*4`
9. `(5 * 2+2)`
10. `(* 5 (2+2))`
11. `(* 5 (2 + 2))`
12. `(5*2 + 2)`
13. `(5*4 + 2)`
14. `(5* (+ 2 2))`
15. `((+ 2 3))`
16. `(/ 6 2)`
17. `/`
18. `(define $ /)`
19. `(define / (* 2 3))`
20. `(/ 6 2)`
21. `/`
22. `($ 6 2)`
23. `$`
24. `(* 2 /)`
25. `(define (double x) (+ x x))`
26. `(double (double 5))`
27. `(define (five) 5)`
28. `(define four 4)`
29. `four`
30. `five`
31. `(four)`
32. `(five)`

2.2 Omgevingsmodellen voor eenvoudige expressies

Doe nu hetzelfde voor de volgende expressies en maak daarenboven gebruik van omgevingsmodellen om de resultaten te bespreken.

1. `(define $ /)`
2. `(define / (* 2 3))`
3. `(/ 6 2)`
4. `/`
5. `($ 6 2)`
6. `$`
7. `(* 2 /)`
8. `(define (double x) (+ x x))`
9. `(double (double 5))`
10. `(define (five) 5)`
11. `(define four 4)`
12. `four`
13. `five`
14. `(four)`
15. `(five)`

3 Syntax van procedures

3.1 fourth

Schrijf een procedure `fourth` die de vierde macht van zijn argument berekent. Doe dit op twee verschillende manieren, eerst eens door gebruik te maken van de vermenigvuldiging en dan eens door `square` te gebruiken.

```
(define (square x) (* x x))
```

3.2 sum-3-squares

Schrijf een procedure `sum-3-squares` die drie getallen als argumenten neemt en de som van hun kwadraten teruggeeft.

3.3 Celsius naar Fahrenheit

Schrijf een procedure `convert-C-to-F` die een temperatuur in graden Celsius omzet in graden Fahrenheit. Gebruik hiervoor de formule: $F = (C + 40) \times 1.8 - 40$.

3.4 Fahrenheit naar Celsius

Schrijf de inverse procedure `convert-F-to-C` die een temperatuur in graden Fahrenheit omzet in graden Celsius.

3.5 oppervlakte en omtrek

Schrijf procedures die de oppervlakte en omtrek van volgende figuren berekenen: driehoek, vierkant, en cirkel.

3.6 oppervlakte en inhoud

Doe hetzelfde voor oppervlakte en inhoud van enkele drie-dimensionale figuren zoals een balk, een bol en een cilinder.

4 Evaluatie van procedures (oefening 4.4 uit Simply Scheme)

Elk van de volgende procedure definities bevat een bepaalde fout. Geef aan wat fout is en corrigeer.

- `(define (sphere-volume r) (* (/ 4 3) 3.14) (* r r r))`
- `(define (next x) (x + 1))`
- `(define (square) (* x x))`
- `(define (triangle-area triangle) (* 0.5 base height))`
- `(define (sum-of-squares (square x) (square y)) (+ (square x) (square y)))`

5 Omgevingsmodellen

Teken de omgevingsmodel-diagrammen voor de evaluatie van enerzijds `(square 5)` en anderzijds `(f 5)` in de veronderstelling dat de volgende procedure-definities gegeven zijn:

```
(define (square x)
  (* x x))

(define (sum-of-squares x y)
  (+ (square x) (square y)))

(define (f a)
  (sum-of-squares (+ a 1) (* a 2)))
```

6 Statische versus dynamische scoping

6.1 omgevingsmodel-diagram

Teken het omgevingsmodel-diagram voor de volgende definities:

- `(define c 3)`
- `(define (add-c x y)
 (+ x y c))`
- `(define (sum x y c)(add-c x y))`

6.2 Voorspel het resultaat van "sum"

Voorspel het resultaat van volgende expressie aan de hand van omgevingsmodel-diagrammen.

```
(sum 1 2 6)
```

6.3 Dynamische scoping

Wat zou het resultaat van `(sum 1 2 6)` geweest zijn indien Scheme een taal met dynamische (i.p.v. statische) scoping was?

7 Definitie van procedures (oefening 4.8 uit Simply Scheme)

7.1 Wetenschappelijke Notatie

De wetenschappelijke notatie van een getal is een manier om zeer grote of zeer kleine getallen weer te geven als een aantal malen een gehele macht van 10. Bijvoorbeeld, 5×10^7 representeert het getal 50000000, en 3.26×10^{-9} drukt het getal 0.00000000326 uit in wetenschappelijke notatie. Schrijf een procedure `(scientific getal exponent)` met twee argumenten: een getal en een exponent van 10 die de corresponderende waarde teruggeeft:

Je kan hierbij gebruik maken van de procedure `(expt n e)` die n^e uitrekent. Merk op dat sommige versies van Scheme breuken als `a/b` weergeen. Andere versies gebruiken de wetenschappelijke notatie. Dus het laatste voorbeeld kan gerepresenteerd worden als "21/5000" of "4.2E-4" in plaats van 0.00042.

```
> (scientific 7 3)
7000.0
> (scientific 42 -5)
0.00042
```

7.2 Een stapje moeilijker

Kan je de procedures schrijven die de andere richting berekenen: Je kan gebruik maken van de primitieve procedures `log` en `floor`.

```
> (sci-coefficient 7000)
7.0
> (sci-exponent 7000)
3.0
```

8 Nog meer definities van procedures (oefening 4.9 en 4.10 uit Simply Scheme)

8.1 Discount

Schrijf een procedure (`discount prijs korting`) die twee argumenten heeft: de initiële prijs van een item en het kortingspercentage. De procedure moet de nieuwe prijs teruggeven:

```
> (discount 10 5)
9.5
> (discount 29.9 50)
14.95
```

8.2 Fooi (tip)

Schrijf een procedure (`tip bedrag`) om de fooi te berekenen die je in een restaurant zou geven. Deze procedure neemt als argument het totaalbedrag en geeft het fooibedrag terug. De fooi is 15% van het totale bedrag maar de procedure moet kunnen afronden zodanig dat het totale geldbedrag (dus de originele prijs plus fooi) een geheel aantal Euro's is. (Gebruik de procedure `ceiling` om af te ronden.)

```
> (tip 19.98)
3.0199999999999996
> (tip 29.23)
4.77
> (tip 7.54)
1.46
```