



- ▶ Introduction and overview
- ▶ Basic types, definitions and functions
- ▶ Basic data structures
- ▶ More advanced data structures
- ▼ Higher order functions

Table of Contents

Functional Expressions

Week 4 Échéance le déc 12, 2016 at 23:30 UTC

Functions as First-Class Values

Week 4 Échéance le déc 12, 2016 at 23:30 UTC

Functions with Multiple Arguments

Week 4 Échéance le déc 12, 2016 at 23:30 UTC

Partial Function Application

Week 4 Échéance le déc 12, 2016 at 23:30 UTC

Mapping Functions on Lists

Week 4 Échéance le déc 12, 2016 at 23:30 UTC

Folding Functions on Lists

Week 4 Échéance le déc 12, 2016 at 23:30 UTC

- ▶ Exceptions, input/output and imperative constructs
- ▶ Modules and data abstraction

OPTIMIZING PARTIAL APPLICATIONS (35/35 points)

Every triangle has a circumscribed circle, that is a circle that goes through the three points of a given triangle. Trigonometry tells us that the radius of this circle is

$$\frac{s}{2 \cdot \cos\left(\frac{a}{2}\right) \cdot 2 \cdot \cos\left(\frac{b}{2}\right) \cdot 2 \cdot \cos\left(\frac{c}{2}\right)} \quad \text{where } a, b \text{ and } c \text{ are the angles of the triangle, and } s \text{ is its circumference.}$$

1. Define a function `ccr: float -> float -> float -> float -> float` that takes as arguments `a`, `b`, `c` and `s`, and returns the radius of circumscribed circle as described above.
2. Update `ccr` so that it does as much work as possible when partially applied to each argument, and minimizes the total number of operations (multiplications, divisions and calls to `cos`).

YOUR OCAML ENVIRONMENT

```
1 let ccr = fun a -> fun b -> fun c -> fun s ->
2   s /. (2. *. cos(a /. 2.) *. 2. *. cos(b /. 2.) *. 2. *. cos(c /. 2.))
3 ;;
4
5 let ccr = fun a ->
6   let a = 8. *. cos(a /. 2.) in fun b ->
7     let b = a *. cos(b /. 2.) in fun c ->
8       let c = b *. cos(c /. 2.) in fun s ->
9         s /. c
10  ;;|
```

Evaluate >

Switch >>

Typecheck

Reset Templ

Full-screen |

Check & Sa

Exercise complete (click for details)

35 pts

Completed, 10 pts

v Exercise 1: ccr

Found ccr with compatible type.

Computing

ccr

```
-1.09410857738563116
3.07754429117092698
-1.7680351079418033
-3.7336916699282261
```

Correct value -26.917383485563068

1 pt

Computing

ccr

```
2.56718390956078046
-3.86732979417683342
-2.16161825761612381
2.71643510296323143
```

Correct value -7.17568985514052216

1 pt

Computing

ccr

```
0.205738145972663489
-4.3800022108363752
2.72858204265634452
-3.8591228431332274
```

Correct value 4.07513676332467512

1 pt

Computing

ccr

```
2.98781369356990201
-2.28344009435281725
```

Computing ccr 3.40480002100653678 -0.775432849929448 0.961430033438616327 2.5331518591175044	
Correct value -2.93963118905272891	1 pt
Computing ccr 0.455280793458246791 1.03715993779053051 -1.84886347232156956 -1.90487282430128246	
Correct value -0.467240994676484389	1 pt
Computing ccr 2.60769308823409229 -1.75237916085929157 4.64686535187950156 0.569830323130857685	
Correct value -0.617134531257467	1 pt
Computing ccr 3.73753200461501756 3.15591415464328939 -0.655583183532410629 3.50604047492054427	
Correct value 220.195174613972682	1 pt
Computing ccr -4.88174113820122191 2.02832215964536822 4.91118693612498447 -3.27690789918453707	
Correct value -1.3108058115237895	1 pt
Computing ccr -4.60455182878205704 -2.68206639526676582 3.52077215060565507 3.09859013579413656	
Correct value 13.5099520111703537	1 pt
v Exercice 2: partial applications	Completed, 25 pts
Found ccr with compatible type.	
The partial application of a is optimized.	5 pts
The partial application of b is optimized.	5 pts
The partial application of c is optimized.	5 pts
The final application of s is optimized.	5 pts
Well done, you made it down to 10 operations.	5 pts

[A propos](#)

[Aide](#)

[Contact](#)

[Conditions générales d'utilisation](#)

[Charte utilisateurs](#)

[Politique de confidentialité](#)

[Mentions légales](#)