

Compilation

`http://www.iro.umontreal.ca/~monnier/6232`

- *Advanced Compiler Design & Implementation* de Steven Muchnick,
- *Building an Optimizing Compiler* de Bob Morgan,
- *Modern Compiler Implementation in ML* de Andrew Appel,
- *Types and Programming Languages* de Benjamin C. Pierce.

Définition de langages

- Règles lexicales, syntaxiques, de typage et d'évaluation

Techniques d'interprétation

- Analyse syntaxique, interprètes, gestion mémoire

Techniques de compilation

- Gestion de la pile, allocation de registres, génération de code
- Analyses de code, représentations internes, optimisations
- Types, objets, fermetures, continuations, interface avec GC
- Compilation de langages dynamiques, JIT compilation

Exemple de compilation -1- Source

```
let fun map f [] = []  
    | map f (x::xs) = (f x) :: (map f xs)  
in map inc  
end
```

Exemple de compilation -2- Inférence de type

```
let map : ('a -> 'b) -> 'a list -> 'b list
  = fn f : 'a -> 'b =>
      fn l : 'a list =>
          case l
            of [] => [] : 'b list
              | (x::xs) => (f x)::(map f xs)
in map inc : int list -> int list
```

Exemple de compilation -3- Filtrage

```
let map =  
  fn f =>  
    fn l =>  
      switch l  
      of [] => []  
         | :: => let x = hd l  
                  xs = tl l  
                  in (f x) :: (map f xs)  
in map inc
```

Exemple de compilation -4- A-normalization

```
map = fn f => fn l =>
  switch l
  of [] => []
  | :: => x = hd l;
        xs = tl l;
        t1 = f x;
        t2 = map f;
        t3 = t2 xs;
        t4 = t1::t3;
        return t4;

t5 = map inc;
return t5
```

Exemple de compilation -5- FixFix

```
map1 = fn (f, l) =>
  switch l
  of [] => []
   | :: => x  = hd l;
        xs = tl l;
        t1 = f x;
        t2 = map f;
        t3 = t2 xs;
        t4 = t1::t3;
        return t4;

map = fn f => fn l => map1 (f, l);
t5 = map inc;
return t5
```

Exemple de compilation -6- Contract

```
map1 = fn (f, l) =>
  switch l
  of [] => []
   | :: => x  = hd l;
           xs = tl l;
           t1 = f x;
           t3 = map1 (f, xs);
           t4 = t1::t3;
           return t4;
t5 = fn l => return map1 (inc, l);
return t5
```


Exemple de compilation -7- CPS

```
map1 = fn (f, l, k1) =>
  switch l
  of [] => k1 []
     | :: => x = hd l;
           xs = tl l;
           k2 = fn t1 =>
                 k3 = fn t3 =>
                       t4 = t1::t3;
                       call k1 (t4);
                       call map1 (f, xs, k3);
                 call f (x, k2);
  t5 = fn (l, k4) => call map1 (inc, l, k4);
  call k (t5)
```

Exemple de compilation -8- Fermetures

```

map1 = fn (f, l, k1) => switch l
  of [] => k1c = k1.0; call k1c ([], k1)
  | :: => x = hd l; xs = tl l;
        k2c = fn (t1, c1) =>
          f = c1.1; xs = c1.2; k1 = c1.3;
          k3c = fn (t3, c2) =>
            t1 = c2.1; k1 = c2.2;
            t4 = t1::t3;
            k1c = k1.0;
            call k1c (t4, k1);
          k3 = vector (k3c, t1, k1);
          call map1 (f, xs, k3);
        k2 = vector (k2c, f, xs, k1);
        call f (x, k2);
t5 = fn (l, k4) => call map1 (inc, l, k4);
call k (t5)
  
```

Exemple de compilation -9- Hoisting (1)

```
map1 (f, l, k1) =  
  switch l  
  of [] => k1c = k1.0; call k1c ([], k1)  
  | :: => x = hd l;  
        xs = tl l;  
        k2 = vector (k2c, f, xs, k1);  
        call f (x, k2);  
  
k2c (t1, c1) =  
  f = c1.1;  
  xs = c1.2;  
  k1 = c1.3;  
  k3 = vector (k3c, t1, k1);  
  call map1 (f, xs, k3);
```

Exemple de compilation -9- Hoisting (2)

```
k3c (t3, c2) =  
  t1 = c2.1;  
  k1 = c2.2;  
  t4 = t1::t3;  
  k1c = k1.0;  
  call k1c (t4, k1);  
  
t5 (l, k4) = call map1 (inc, l, k4);  
  
call k (t5)
```

Exemple de compilation -10- CodeGen (1)

```
map1:   mv f, r1           #  
        mv l, r2      #  
        mv k1, r3     # fn (f, l, k1) =>  
        bnz l, case2  # switch l {.. of [] ..}  
        ld k1c, k1(0) # k1c = k1.0  
        mv r1, 0      #  
        mv r2, k1     #  
        jmp k1c       # call k1c ([], k1)
```

Exemple de compilation -10- CodeGen (2)

```
case2:                # | :: =>
    ld x, l(0)         # x = hd l
    ld xs, l(4)        # xs = t1 l
    sub sp, sp, 16    #
    st sp(12), k1      #
    st sp(8), xs       #
    st sp(4), f        #
    st sp(0), k2c      #
    mv k2, sp          # k2 = vector (k2c, f, xs, k1)
    mv r1, x           #
    mv r2, k2          #
    jmp f              # call f (x, k2)
```

Exemple de compilation -10- CodeGen (3)

```
k2c:    mv t1, r1          #
        mv c1, r2     # fn (t1, c1) =>
        ld f, c1(4)   # f = c1.1
        ld xs, c1(8)  # xs = c1.2
        ld k1, c1(12) # k1 = c1.3
        sub sp, sp, 12 #
        st sp(8), k1   #
        st sp(4), t1   #
        st sp(0), k3c  #
        mv k3, sp      # k3 = vector (k3c, t1, k1)
        mv r1, f       #
        mv r2, xs      #
        mv r3, k3      #
        jmp map1       # call map1 (f, xs, k3)
```

Exemple de compilation -10- CodeGen (4)

```
k3c:    mv t3, r1           #  
        mv c2, r2     # fn (t3, c2) =>  
        ld t1, c2(4)  # t1 = c2.1  
        ld k1, c2(8)  # k1 = c2.2  
        sub sp, sp, 8 #  
        st sp(4), t3  #  
        st sp(0), t1  #  
        mv t4, sp     # t4 = t1::t3  
        ld k1c, k1(0) # k1c = k1.0  
        mv r1, t4     #  
        mv r2, k1     #  
        jmp k1c       # call k1c (t4, k1)
```


Exemple de compilation -10- CodeGen (5)

```
t5:      mv l, r1          #  
        mv k4, r2      # fn (l, k4) =>  
        mv r1, inc     #  
        mv r2, l       #  
        mv r3, k4      #  
        jmp map1       # call map1 (inc, l, k4)
```

Exemple de compilation -11- RegAlloc (1)

```
map1:   mv r4, r1           # fn (f, l, k1) =>
        bnz r2, case2  # switch l {.. of [] ..}
        ld r4, r3(0)   # k1c = k1.0
        mv r1, 0       #
        mv r2, r3       #
        jmp r4          # call k1c ([], k1)
```

Exemple de compilation -11- RegAlloc (2)

```

case2:                                # | :: =>
    ld r1, r2(0)                        # x = hd l
    ld r2, r2(4)                        # xs = t1 l
    sub sp, sp, 16                       #
    st sp(12), r3                        #
    st sp(8), r2                         #
    st sp(4), r4                         #
    st sp(0), k2c                        #
    mv r2, sp                            # k2 = vector (k2c, f, xs, k1)
    jmp r4                               # call f (x, k2)
  
```

Exemple de compilation -11- RegAlloc (3)

```

k2c:                # fn (t1, c1) =>
                    # f = c1.1
    ld r3, r2(4)     # xs = c1.2
    ld r4, r2(8)     # k1 = c1.3
    ld r2, r2(12)    #
    sub sp, sp, 12  #
    st sp(8), r2     #
    st sp(4), r1     #
    st sp(0), k3c    #
    mv r5, sp        # k3 = vector (k3c, t1, k1)
    mv r1, r3        #
    mv r2, r4        #
    mv r3, r5        #
    jmp map1         # call map1 (f, xs, k3)
  
```

Exemple de compilation -11- RegAlloc (4)

```

k3c:                                # fn (t3, c2) =>
    ld r3, r2(4)                     # t1 = c2.1
    ld r2, r2(8)                     # k1 = c2.2
    sub sp, sp, 8                    #
    st sp(4), r1                     #
    st sp(0), r3                     #
    mv r1, sp                        # t4 = t1::t3
    ld r3, r2(0)                     # k1c = k1.0
    jmp r3                           # call k1c (t4, k1)
  
```

Exemple de compilation -11- RegAlloc (5)

```
t5:      mv r4, r1          #  
        mv r3, r2      # fn (l, k4) =>  
        mv r1, inc     #  
        mv r2, r4      #  
        jmp map1       # call map1 (inc, l, k4)
```