

# Cervoise: Draft 03/2015

Jacques-Pascal Deplaix

## Grammar before March 2015:

```
toplevel-expr ::= 'let' lower-name argument* (':' type)? '=' term
               | 'let' 'rec' lower-name argument* ':' type '=' term
               | 'type' 'alias' upper-name '=' type
               | 'type' upper-name-ident (':' kind)? '=' '|' variants
               | 'exception' upper-name-ident type*
               | 'let' lower-name ':' type '=' 'begin' <llvm-code> 'end'

lower-name ::= ['a' 'z'] (['a' 'z'] | ['A' 'Z'])*
upper-name ::= ['A' 'Z'] (['a' 'z'] | ['A' 'Z'])*
module-name ::= upper-name ( '.' upper-name ) *
lower-ident ::= (module-name '.')? lower-name
upper-ident ::= (module-name '.')? upper-name

variants ::= upper-name ':' type ( '|' variants ) ?

argument ::= upper-name
           | '(' lower-name ':' type ') '

type-argument ::= upper-name
                | '(' upper-name ':' kind ') '

kind ::= '*'
       | kind '->' kind
       | '(' kind ') '

exns ::= upper-ident ( '|' exns ) *

type ::= upper-ident
       | type '->' type
       | type '-[' exns ']->' type
       | 'forall' type-argument+ ',' type
       | 'λ' type-argument+ ',' type
       | type type
       | '(' type ') '

term ::= lower-ident
       | 'λ' argument+ '->' term
       | 'let' lower-name argument* (':' type)? '=' term 'in' term
       | 'let' 'rec' lower-name argument* ':' type '=' term 'in' term
       | term term
       | term '[' type ']'
       | 'fail' '[' type ']' exn-value
       | 'match' term 'with' '|' ? pattern-clauses 'end'
       | 'try' term 'with' '|' ? try-patterns 'end'
       | '(' term ') '

exn-value ::= upper-name
           | '(' upper-name term+ ') '

pattern-clauses ::= upper-ident pattern-args* '->' term ( '|' pattern-clauses ) ?

pattern-args ::= lower-name
              | upper-ident
              | '(' upper-ident pattern-args+ ') '

try-pattern-clauses ::= upper-ident lower-name* '->' term ( '|' try-pattern-clauses ) ?
```

