Software Verification

Lecture 02. OCaml Programming I

Ji, Yong-Hyeon

24. 07. 18 (Thu)

Coding & Optimization Together (CO2)

Crypto & Security Engineering Lab (CSE)

Department of Information Security, Cryptology, and Mathematics

Table of Contents

1. Motivation

- 2. Basic OCaml Programming
 - 2.1 OCaml 기본 구성

3. Advanced OCaml Programming

Motivation

1. Motivation

https://www.tiobe.com/tiobe-index/

| Jul 2024 | Jul 2023 | Change | Program | nming Language | Ratings | Change |
|----------|----------|--------|-----------|----------------------|---------|--------|
| 1 | 1 | | • | Python | 16.12% | +2.70% |
| 2 | 3 | ^ | 0 | C++ | 10.34% | -0.46% |
| 3 | 2 | • | Θ | С | 9.48% | -2.08% |
| 4 | 4 | | <u>\$</u> | Java | 8.59% | -1.91% |
| 5 | 5 | | 0 | C# | 6.72% | -0.15% |
| 6 | 6 | | JS | JavaScript | 3.79% | +0.68% |
| 7 | 13 | * | -60 | Go | 2.19% | +1.12% |
| 8 | 7 | • | VB | Visual Basic | 2.08% | -0.82% |
| 9 | 11 | ^ | B | Fortran | 2.05% | +0.80% |
| 10 | 8 | • | SQL | SQL | 2.04% | +0.57% |
| 11 | 15 | * | (3) | Delphi/Object Pascal | 1.89% | +0.91% |
| 12 | 10 | • | ♠ | MATLAB | 1.34% | +0.08% |
| 13 | 17 | * | ® | Rust | 1.18% | +0.29% |
| 14 | 16 | ^ | 4 | Ruby | 1.16% | +0.25% |
| 15 | 12 | • | | Scratch | 1.15% | +0.08% |
| 16 | 9 | ¥ | php | PHP | 1.15% | -0.27% |
| 17 | 18 | ^ | 8 | Swift | 1.13% | +0.25% |
| 18 | 14 | ¥ | ASM | Assembly language | 1.11% | +0.10% |
| 19 | 20 | ^ | - | COBOL | 1.08% | +0.21% |
| 20 | 26 | * | • | Kotlin | 1.05% | +0.35% |
| | | | | | | |

1. Motivation

| 22 R 0.83% 23 SAS 0.79% 24 Ada 0.79% 25 Dut 0.74% 26 D 0.72% 27 Usp 0.67% 28 Prolog 0.67% 29 0.67% 0.66% 30 Perl 0.66% 31 Haskal 0.66% 32 Un 0.60% 33 Scola 0.59% 34 Jula 0.69% 35 Olfective C 0.40% 36 VSScript 0.40% 37 GAMS 0.33% 38 ML 0.31% 39 Solidry 0.40% 39 Solidry 0.20% 40 Loga 0.20% 41 Deposited 0.20% 42 Tannack SQL 0.20% 43 PoweStell 0.20% 44 PoweStell 0.20% <th>Position</th> <th>Programming Language</th> <th>Ratings</th> | Position | Programming Language | Ratings |
|---|----------|----------------------|---------|
| 23 SAS 0,79% 24 Ada 0,79% 25 Dat 0,72% 26 Lip 0,72% 27 Lip 0,67% 28 Prolog 0,67% 29 (Mosul) FoxPro 0,60% 30 Perl 0,60% 31 Haskell 0,60% 32 Lia 0,60% 33 Scala 0,50% 34 Julia 0,50% 35 Olgerewe C 0,40% 37 GAS 0,33% 37 GAMS 0,33% 38 M 0,20% 40 Logo 0,20% 40 Logo 0,20% 41 PLSQ 0,20% 42 Tanoaci SQL 0,27% 43 PowerShell 0,20% 44 PowerShell 0,20% 45 Bash 0,20% 46 Scheme 0,20% | 21 | Classic Visual Basic | 0.91% |
| 24 Ada 0.78% 25 Dut 0.74% 26 D 0.72% 27 Unp 0.67% 28 Prolog 0.67% 29 (Mount Food Prolog) 0.66% 30 Perl 0.66% 31 Lis 0.66% 32 Lis 0.69% 34 Julia 0.59% 35 Olegeber C 0.69% 36 VBS-rigt 0.40% 37 GAMS 0.33% 38 M. 0.33% 39 Solidty 0.30% 40 Lopo 0.29% 41 Lips 0.29% 42 PLSCI 0.29% 43 PowerStell 0.27% 44 PowerStell 0.27% 45 Bash 0.28% 46 ABAP 0.24% 47 ABAP 0.24% | 22 | R | 0.83% |
| 25 Dart 0.74% 26 D 0.72% 27 Lisp 0.67% 28 Prolog 0.67% 29 (Mossa) FooPro 0.66% 30 Perl 0.66% 31 Hasskell 0.66% 32 Lisa 0.60% 33 Scole 0.59% 34 Julia 0.59% 35 Citylective C 0.40% 36 VSScript 0.40% 37 GMS 0.33% 38 ML 0.31% 39 Solidity 0.30% 40 Ligo 0.20% 41 Ligo 0.20% 42 Tannack SQL 0.27% 43 PowerStreit 0.20% 44 TypeScript 0.20% 45 Sachene 0.20% 46 Scheme 0.20% 47 Alake 0.24% 48 Alake | 23 | SAS | 0.79% |
| 26 D 0,72% 27 Lisp 0,67% 28 Polog 0,67% 29 (Mossil Forlin) 0,66% 30 Perl 0,66% 31 Haskell 0,66% 32 Lia 0,66% 34 Julia 0,59% 34 Julia 0,56% 36 OBjective C 0,40% 37 GAMS 0,33% 38 ML 0,31% 39 Solidry 0,30% 40 Lio 0,20% 40 Lio 0,20% 41 PL/SQL 0,20% 42 Tanosat SQL 0,27% 43 PowerShell 0,20% 44 PyesScript 0,20% 45 Bash 0,20% 46 Scheme 0,20% 47 Ala Ala 0,20% | 24 | Ada | 0.78% |
| 27 Lisp 0.67% 28 Prolog 0.67% 29 (Mosal) GodPo 0.66% 30 Perl 0.66% 31 Haisafel 0.66% 32 Lia 0.59% 34 Julia 0.59% 35 Olectwick 0.40% 36 VBScript 0.40% 37 GAMS 0.33% 38 M. 0.33% 39 Mc 0.33% 30 Solidity 0.30% 40 Lopo 0.20% 41 Ego 0.20% 42 Tannact-SQL 0.27% 43 PowerStell 0.27% 44 Pyestirpl 0.26% 45 Bash 0.28% 46 Scheme 0.24% 47 ABAP 0.24% | 25 | Dart | 0.74% |
| 28 Polog 6.87% 29 (Mosup) FoxPro 6.66% 31 Haskell 0.66% 32 Lu 0.60% 32 Scale 0.50% 34 Jula 0.56% 35 Cigedive C 0.40% 36 VSScript 0.40% 37 GAMS 0.31% 38 ML 0.31% 39 Solidity 0.20% 44 Logo 0.20% 42 Tannack SQL 0.27% 43 PowerStell 0.27% 44 TypeScript 0.20% 45 Scheme 0.20% 46 Scheme 0.20% 47 ABAP 0.24% | 26 | D | 0.72% |
| 29 (Young) Fou ^{Ph} o 0.66% 30 Perl 0.66% 31 Haskell 0.66% 32 Lua 0.60% 33 Scola 0.59% 34 Julia 0.59% 35 Objective C 0.40% 36 VSSorje 0.40% 37 GAMS 0.31% 39 ML 0.31% 30 Solidry 0.50% 40 Lop 0.20% 41 PUSQL 0.20% 42 Tansact SQL 0.27% 43 PowerShell 0.20% 44 TypeScipt 0.20% 45 Bash 0.20% 46 Scheme 0.20% 47 AlAP 0.24% 48 Auk 0.22% | 27 | Lisp | 0.67% |
| 30 Perf 0.66% 31 Haisafal 0.65% 32 Lia 0.59% 33 Scala 0.59% 34 Jula 0.56% 35 Olgerback 0.40% 37 GAMS 0.33% 38 M. 0.33% 39 Mc 0.30% 40 Lopo 0.20% 41 PUSQ 0.20% 42 Tannact-SQL 0.27% 43 PowerStell 0.27% 44 PowerStell 0.29% 45 Bash 0.28% 46 Scheme 0.24% 47 AAR 0.44 0.24% 46 AAR 0.24% | 28 | Prolog | 0.67% |
| 31 Plaskell 0.65% 32 Lua 0.60% 34 Julia 0.59% 35 Olgetüve C 0.40% 37 GAMS 0.33% 38 ML 0.31% 39 Solidby 0.30% 44 Logo 0.29% 42 PUSQL 0.25% 42 Tannact SQL 0.27% 43 PowerStell 0.27% 44 TypeScipt 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 46 ABAP 0.24% | 29 | (Visual) FoxPro | 0.66% |
| 32 Lia 0.60% 33 Scala 0.59% 34 Julia 0.50% 35 Objective C 0.40% 36 VBS-rigt 0.40% 37 GAMS 0.33% 38 M. 0.31% 39 Soliday 0.00% 40 Lopo 0.20% 41 PUSQL 0.20% 42 Tannack SQL 0.27% 43 PowerShell 0.20% 44 TypeScipt 0.20% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 Auk 0.22% | 30 | Perl | 0.66% |
| 33 Scala 0.59% 34 Julia 0.50% 35 Olgerbac C 0.40% 36 VBScript 0.40% 37 GAMS 0.33% 38 M. 0.30% 40 Logo 0.20% 41 PUSQ 0.20% 42 Tansact SQL 0.27% 43 PowerStrell 0.27% 44 TypeScript 0.29% 45 Bash 0.28% 46 Scheme 0.24% 47 AAR 0.24% 48 AaK 0.22% | 31 | Haskell | 0.65% |
| 34 Julia 0.56% 35 Objective C 0.40% 36 VBSorpt 0.40% 37 GAMS 0.33% 38 ML 0.31% 39 Solidhy 0.30% 40 Lope 0.20% 41 PLSQL 0.28% 42 Transact-SQL 0.27% 43 PowerShell 0.27% 44 TypeScript 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 0.22% | 32 | Lua | 0.60% |
| 35 Clylective C 0.40% 36 VBSropt 0.40% 37 GAMS 0.33% 38 ML 0.31% 39 Soliday 0.20% 40 Logo 0.20% 41 PUSQL 0.20% 42 Tannact SQL 0.27% 43 PowerSnell 0.27% 44 TypeScipt 0.20% 45 Bash 0.20% 46 Scheme 0.24% 47 ABAP 0.24% 48 Auk 0.22% | 33 | Scala | 0.59% |
| 36 VBSorjat 0.40% 37 GAMS 0.33% 38 M. 0.31% 30 Suldity 0.30% 40 Lopo 0.20% 41 PUSQL 0.20% 42 Tansact-SQL 0.27% 43 PowerShell 0.27% 44 TypeScript 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 46 Ask 0.22% | 34 | Julia | 0.56% |
| 37 GAMS 0.33% 38 ML 0.31% 39 Solidry 0.30% 40 Logo 0.28% 41 PLSQL 0.28% 42 Transact-SQL 0.27% 43 PowerShell 0.27% 44 TypeScipt 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 Auk 0.22% | 35 | Objective-C | 0.40% |
| 38 ML 0.31% 39 Soliday 0.30% 40 Logo 0.28% 41 PUSQL 0.28% 42 Tannack SQL 0.27% 43 PowerShell 0.27% 44 TypeScipt 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 Ank 0.22% | 36 | VBScript | 0.40% |
| 39 Solidity 0.30% 40 Logo 0.29% 41 PLSQL 0.29% 42 Tannact-SQL 0.27% 43 PowerSrell 0.27% 44 TypeScript 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 Auk 0.22% | 37 | GAMS | 0.33% |
| 40 Logo 0.28% 41 PLSQL 0.29% 42 Tanoact-SQL 0.27% 43 PowerStell 0.27% 44 Type-Script 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 Auk 0.22% | 38 | ML | 0.31% |
| 41 PLSQL 0.28% 42 Tananat-SQL 0.27% 43 PowerShell 0.27% 44 TypeScript 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 Ausk 0.22% | 39 | Solidity | 0.30% |
| 42 Transact SQL 0.27% 43 PowerShel 0.27% 44 TypeScript 0.26% 45 Bash 0.28% 46 Scheme 0.24% 47 ABAP 0.24% 48 Ank 0.22% | 40 | Logo | 0.28% |
| 43 PowerShell 0.27% 44 TypeScript 0.26% 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 Aak 0.22% | 41 | PL/SQL | 0.28% |
| 44 TypeScript 0,26% 45 Bash 0,20% 46 Scheme 0,24% 47 ABAP 0,22% | 42 | Transact-SQL | 0.27% |
| 45 Bash 0.26% 46 Scheme 0.24% 47 ABAP 0.24% 48 Auk 0.22% | 43 | PowerShell | 0.27% |
| 46 Scheme 0.24% 47 ABAP 0.24% 48 Auk 0.22% | 44 | TypeScript | 0.26% |
| 47 ABAP 0.24% 48 Auk 0.22% | 45 | Bash | 0.26% |
| 48 Ausk 0.22% | 46 | Scheme | 0.24% |
| | 47 | ABAP | 0.24% |
| 49 Ladder Logic 0.22% | 48 | Awk | 0.22% |
| | 49 | Ladder Logic | 0.22% |
| 50 RPG 0.21% | 50 | RPG | 0.21% |

Basic OCaml Programming

OCaml 프로그램의 기본 단위 공식

• 프로그램을 구성하는 두 가지 기본 단위

Statement:

$$x = x + 1$$

Expression:

$$(x + y) * 2$$

▷ OCaml 프로그램의 기본 단위 공식

- 프로그램을 구성하는 두 가지 기본 단위
 - * Statement

$$x = x + 1$$

* Expression

$$(x + y) * 2$$

▷ OCaml 프로그램의 기본 단위 공식

- 프로그램을 구성하는 두 가지 기본 단위
 - * Statement (명령문): 기계 상태를 변경

$$x = x + 1$$

* Expression (식): 상태 변경 없이 값을 계산

$$(x + y) * 2$$

▷ OCaml 프로그램의 기본 단위 공식

- 프로그램을 구성하는 두 가지 기본 단위
 - * Statement (명령문): 기계 상태를 변경

$$x = x + 1$$

* Expression (식): 상태 변경 없이 값을 계산

$$(x + y) * 2$$

- 프로그래밍 언어를 구분하는 한가지 기준:
 - * 명령문을 중심으로 프로그램을 작성
 - C, C++, Java, Python, JavaScript, etc.
 - "Imperative Languages"
 - * 식을 중심으로 프로그램을 작성
 - ML, Haskell, Scala, Lisp, etc.
 - "Functional Programming"

▷ OCaml 프로그램의 기본 구조

• 값 정의들의 나열

$$\begin{array}{c} \text{let } x_1 = e_1 \\ \\ \text{let } x_2 = e_2 \\ \\ \vdots \\ \\ \text{let } x_n = e_n \end{array}$$

- * 식 e_1, e_2, \ldots, e_n 을 순차적으로 계산
- * 변수 x;는 식 e;의 값을 지칭 (assignment vs binding)

```
let hello = "Hello"
let world = "World"
let helloworld = hello ^ " " ^ world
let _ = print_endline helloworld
```

Compile

```
@:∼$>ocaml helloworld.ml
```

OCaml REPL(Real-Eval-Print Loop)

```
@:~$>ocaml
# #use "helloworld.ml";;
val hello : string = "Hello"
val world : string = "World"
val helloworld : string = "Hello World"
Hello World
- : unit = ()
# exit 1;;
```

2.1 OCaml 기본 <u>구성</u>

- ▷ Arithmetic Expression (산술식)
 - 정수와 실수

```
# 1 + 2 * 3;;

- : int = 7

# 1.1 +. 2.2 *. 3.3;;

- : float = 8.36
```

정수값을 위한 산술 연산자

$$+$$
, $-$, $*$, $/$, mod

실수값을 위한 산술 연산자

- ▷ Arithmetic Expressions (산술식)
 - 정수 타입과 실수 타입을 명확히 구분하자

```
# 3 + 2.0;;
Error: This expression has type float but an
expression was expected of type int
# 3 + int_of_float 2.0;;
- : int = 5
```

- ▷ Boolean Expressions (논리식)
 - 논리값

```
# true;;
- : bool = true
# false;;
- : bool = false
```

비교 연산자 (산술식 → 논리식)

```
# 1 = 2;;
- : bool = false
# 1 <> 2;;
- : bool = true
# 2 <= 2;;
- : bool = true</pre>
```

- ▷ Boolean Expressions (논리식)
 - 논리 연산자 (논리식 → 논리식)

```
# true && (false || not false);;
- : bool = true
# (2 > 1) && (3 > 2);;
- : bool = false
```

▷ Primitive Values (기본값)

- OCmal은
 - integer (정수)
 - float (실수)
 - boolean (논리)
 - character (문자)
 - string (문자열)
 - unit (유닛)

을 제공

```
# 'c';;
- : char = 'c'
# "Objective " ^ "Caml";;
- : string = "Objective Caml"
# ();;
- : unit = ()
```

▷ Conditional Expression (조건식)

```
if e_1 then e_2 else e_3
```

```
# if 1 then 2 else 3;;
```

▷ Conditional Expression (조건식)

if e_1 then e_2 else e_3

• *e*₁은 반드시 논리식이어야 함. 즉 *e*₁의 값은 true or false

if 1 then 2 else 3;;
Error: This expression has type int but an
expression was expected of type bool because it is
in the condition of an if-statement

- ▷ Conditional Expression (조건식)
 - 조건식의 값은 e1 값에 따라서 결정

```
# if 2 > 1 then 0 else 1;;
- : int = 0
# if 2 < 1 then 0 else 1;;
- : int = 1</pre>
```

• *e*₂와 *e*₃는 타입이 같아야 함

```
# if true then 1 else true;;
Error: This expression has type bool but an
expression was expected of type int
```

▷ Function Expression (함수식)

$$fun x \rightarrow e$$

- 함수의 예:
 - fun $x \rightarrow x + 1$
 - fun $y \rightarrow y * y$
 - fun $x \rightarrow$ if x > 0 then x + 1 else x * x
 - fun $x \rightarrow$ fun $y \rightarrow x + y$
 - fun $x \rightarrow$ fun $y \rightarrow$ fun $z \rightarrow x + y + z$

Advanced OCaml Programming

To be continue ...