OCamlでらくらく関数型プログラミング

ocaml-nagoya
今井 宜洋

2009年8月22日

http://www.itpl.co.jp/ocaml-nagoya/

OCamlとは:

- Objective Camlというプログラミング言語
- フランスの国立研究所INRIAで研究、開発
- 数学的な理論に基づいて設計された言語

OCamlと名古屋:

- 名大の数学の研究室に開発者の一人がいる
- 勉強コミュニティ ocaml-nagoyaがある
- 名古屋の会社(有)ITプランニング はOCamlで仕事をしている

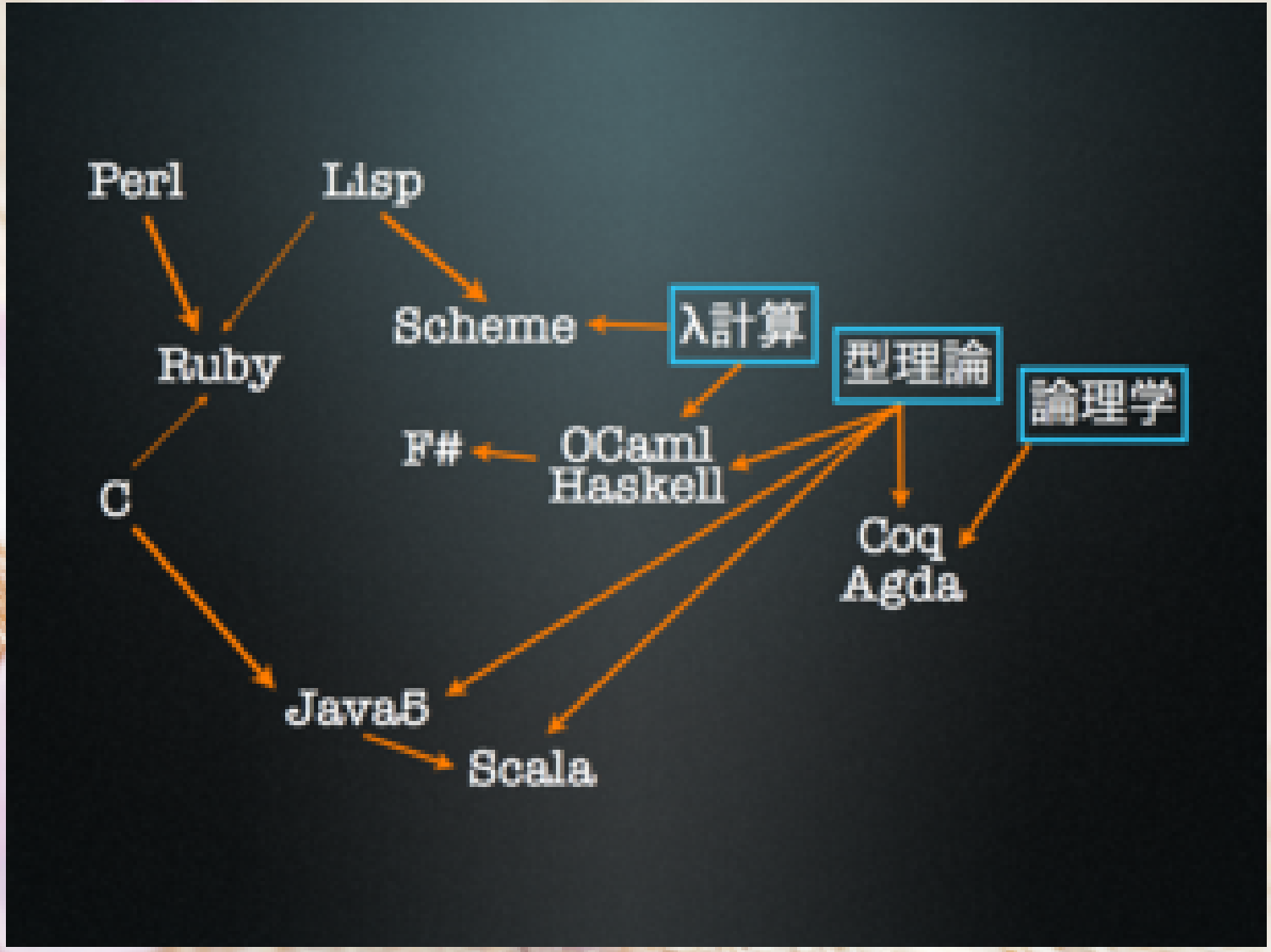
発表の流れ

- OCamlと数学理論
- OCamlの５つの長所
- OCamlはこんなところで使われている
- OCamlの欠点
- まとめ

OCamlと数学の理論

OCamlはプログラミング言語を研究している人が作った。

- λ 計算
- 型理論


Perl
Lisp
Scheme
λ計算
型理論
論理学
Ruby
F#
OCaml
Haskell
C
Coq
Agda
Java5
Scala

OCamlと数学の理論

一言でいうとCやJavaより安全で、
Perl,Rubyみたいにシンプル。

OCamlの５つの長所

- 安全である
- 柔軟
- シンプル
- 実行効率がよい
- 定理証明器と仲良し

OCamlは安全

静的で強い型システム

→ コンパイル時に型チェック

OCamlは安全

例:

```
let register name =
  ...

register "Yoshihiro";
register "Ayumi"
```

OCamlは安全

例:

```
let register age name =
  if age >= 20 then
    ...

register "Yoshihiro";
register "Ayumi"
```

OCamlは安全

例:

```
let register age name =
  if age >= 20 then
    ...

register "Yoshihiro"; (* 型エラー *)
register "Ayumi"      (* 型エラー *)
```

OCamlは安全

例:

```
let register age name =
  if age >= 20 then
    ...

register "Yoshihiro"; (* 型エラー *)
register "Ayumi"      (* 型エラー *)
```

型チェックがないと、関係する全ての部分を目で探して修正しなければならない。

OCamlは安全

代数的データ型とパターンマッチング　→ null参照エラーは起き得ない。

OCaml:

```
match search "imai" with
| Some imai ->
    ...
| None ->
    ...
```

Java:

```
Person imai = search "imai";
if (imai != null) {
    ...
} else {
    ...
}
```

OCamlは安全

問題: リスト(配列)の１番目と２番目を入れ換える

Java:

```
public <T> void swap(ArrayList<T> xs) {
    if (xs.size() >= 2) {
        T tmp = xs.get(0);
        xs.set(0, xs.get(1));
        xs.set(1, tmp);
    }
}
```

OCaml:

```
let swap = function
  | x1 :: x2 :: tail -> x2 :: x1 :: tail
  | l -> l
```

OCamlは柔軟

高階関数

関数が第一級。関数が関数を返したり、クロージャーを渡すことができる。

OCamlは柔軟

高階関数 例: map $[x_1, x_2, .., x_n] \rightarrow [f(x_1), f(x_2), .., f(x_n)]$

OCaml:

```
List.map (fun x -> x * x) [1; 2; 3]
```

ruby:

```
[1, 2, 3].map { |x| x * x }
```

OCamlは柔軟

高階関数 例: iterator

OCaml:

```
List.iter print_string [ "It "; "is "; "cool!"]
```

ruby:

```
[ "It ", "is ", "cool!"].each { |x| puts x }
```

OCamlは柔軟

高階関数 例: db proc

```
let exec proc =
  let conn = DB.open () in
  let result =
    try proc conn with
    | err ->
        DB.close conn;
        raise err
  in
  DB.close conn;
  result
```

OCamlは柔軟

OCamlではループじゃなくて再帰を使う。
再帰関数 例: GCD 最大公約数を求める関数

OCaml:

```
let rec gcd x y =
  if y=0 then x
  else gcd y (x mod y)
```

ruby:

```
def gcd( a, b )
  i = a
  while i > 0
    break if (a % i == 0 && b % i == 0)
    i -= 1
  end
  return i
end
```

OCamlはシンプル

完全な型推論

型を書かなくてもよい。すべての型を省略することができる。

OCamlはシンプル

完全な型推論 例: succ

Java:

```
public static int succ(int x) {
    return x + 1;
}
```

OCaml:

```
let succ x = x + 1
```

OCamlはシンプル

完全な型推論 例: Hash

Java:

```
HashMap<String,ArrayList<Integer>> hash =
    new HashMap<String, ArrayList<Integer>>();
hash.put("Imai",
         new ArrayList(new Integer[]{1,2,3}));
```

OCaml:

```
let hash = Hashtbl.create() in
Hashtbl.set hash "Imai" [1;2;3]
```

Ruby:

```
hash = Hash.new
hash["Imai"] = [1,2,3]
```

OCamlはシンプル

完全な型推論 例: id Java:

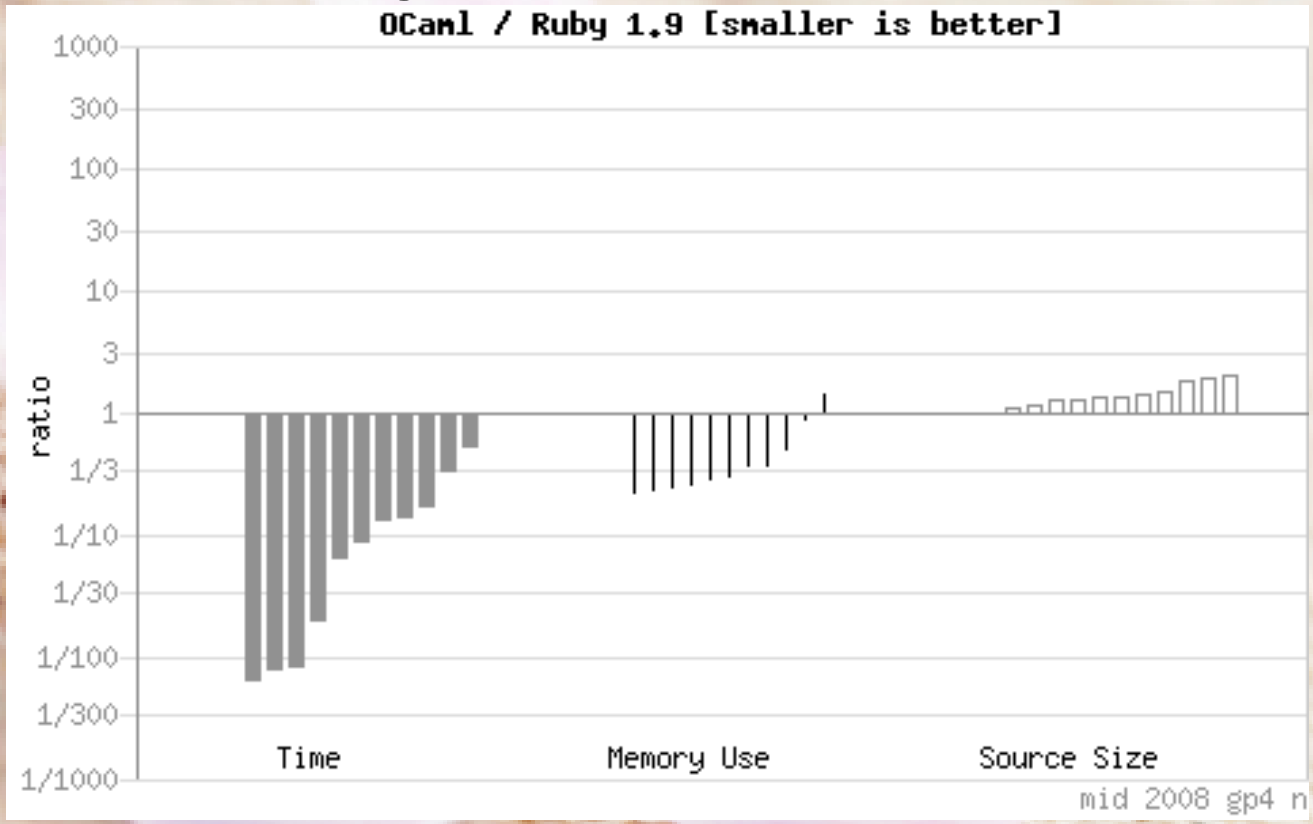
```
public static <T> T id(T x) {
    return x;
}
```

OCaml:

```
let id x = x
```

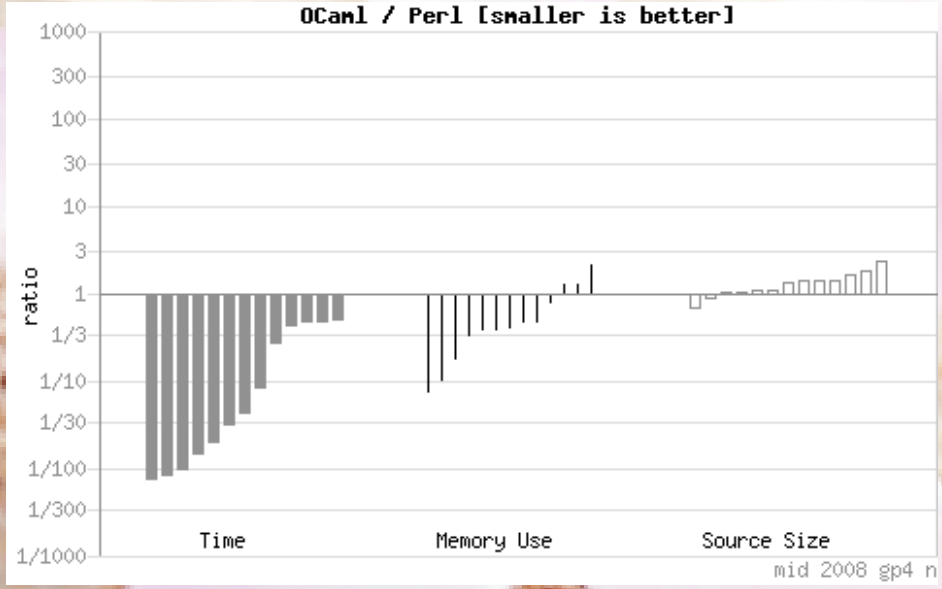
効率がよい

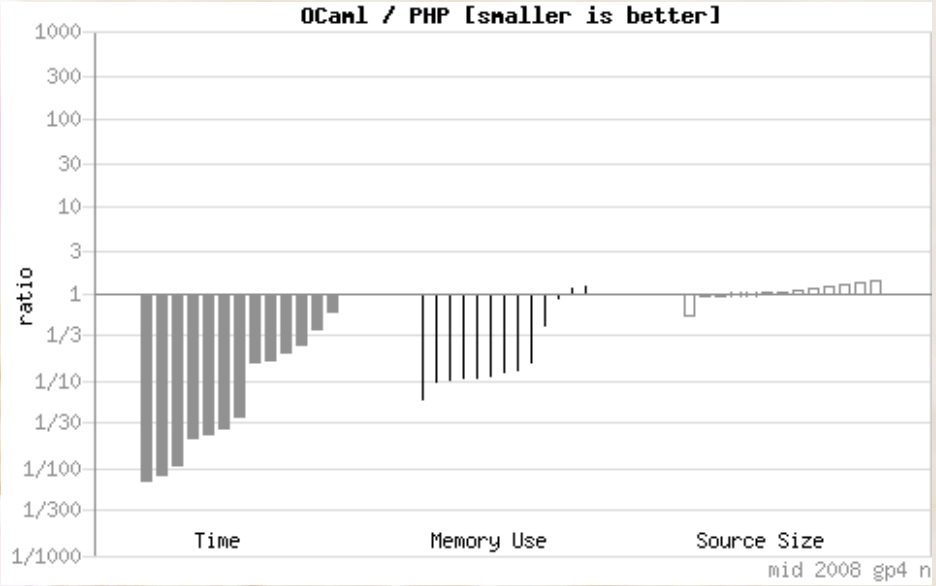
OCamlとRuby 1.9:



効率がよい

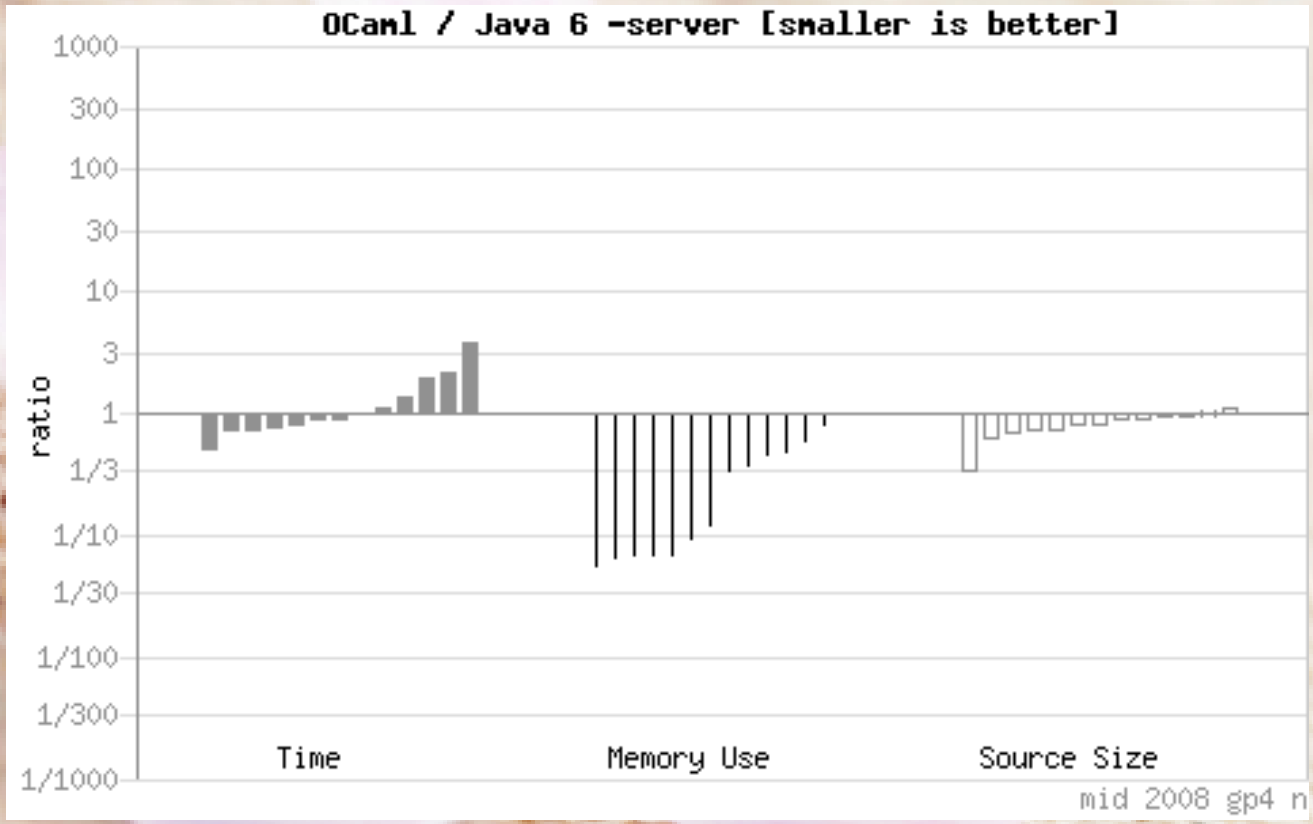
OCamlとPerl,PHP:





効率がよい

OCamlとJava6:



OCamlは定理証明器と仲良し

Coqと仲良し

OCamlは定理証明器と仲良し

定義した関数の性質を述べて保証する事ができる。

- 入力が0より大きいなら戻り値も0より大きい
- エンコード→デコードで必ず元に戻る
- cgiの結果は必ずHTMLの規格に準拠している
- 最大公約数

OCamlは定理証明器と仲良し

OCamlなどのソースコードに変換できる。

- 既存のOCamlのライブラリや自分のプログラムと容易に連携できる
- 逆に誰かの作ったCoqライブラリをOCamlから使える

OCamlの欠点

- ドキュメントが少ない
- 標準ライブラリが少ない

OCamlはこんなところで使われている

- オープンソースプログラム
  - ファイル同期システム unison
  - 世界最速のフーリエ変換
  - tennis game
- 産業
  - 有限会社 IT プランニング
  - Jane Street Capital

OCamlのインストール

Debian，Ubuntuの場合:

```
# apt-get install ocaml
```

その他のLinux:
http://caml.inria.fr/download.htmlからソースコードをダウンロード

```
$ ./configure
$ make world.opt
# make install
```

Mac OS X:

```
# ports install ocaml
```

おわりに

私たちと一緒に型推論しませんか？

ocaml-nagoyaはメンバーを大募集しています。

参加希望者はocaml-nagoya @ freeml.com までメールください。

URL:

- ocaml-nagoya: http://www.itpl.co.jp/ocaml-nagoya/
- OCaml-JP: http://ocaml.jp/

東都大駱駝會

會日己丑歲八月卅日
知東京大學山上會館