

# ここ1、2年の 活動内容

h\_sakurai

# モチベーション

- ・ Cを置き換えられる言語を作りたい
- ・ Scalaは素晴らしいけど、JVMが必要で、コンパイルが遅いのがネック。JVMのないScalaのような言語が欲しい
- ・ Cを置き換えられるGCがない、自由度の高い関数型言語を作りたい
- ・ プログラムでプログラムを操作出来るLispのマクロのような事をしたい

# やっつけたこと

- ・ LLVM用の言語作成とその作り方を書く
- ・ OCamlを出力する言語Camlupの作成
- ・ OCamlでJavaバイトコードの読み込み
- ・ OCamlでのJavaのサブセット言語のリファクタリング
- ・ OCamlによるScalaのパース作成

# やっつけたこと

- ・ OCamlの高速な型推論の翻訳
- ・ Haskell In Typing Haskellの翻訳
- ・ Haskellの型クラスをHaskell, OCamlで実装したもの  
の理解
- ・ LLVMコンパイラをScala, OCamlで作る方法を書く
- ・ min-camlをゼロから作るやり方を書く



# LLong LLVM出力言語

- <https://github.com/hsk/llong/>

```
def main() {  
  var a = 1  
  print_l(a)  
}
```

```
def main() {  
  var a = 1  
  var b = a + 3  
  print_l(b)  
}
```

```
def main() {  
  var a:int = 1  
  print_i(b)  
}
```

```
def main() {  
  var a:Array[int,10]  
  for(var i = 0; i < 10; i++)  
    a[i] = i;  
  for(var i = 0; i < 10; i++)  
    print_i(a[i])  
}
```

```
def main() {  
  var a:struct{a:int b:struct{a:int b:int}}  
  a.b.a = 2  
  print_i(a.b.a)  
}
```

```
def main() {  
  var a:int = 1  
  var p:Ptr[int] = &a  
  *p = *p + 1  
  print_i(a)  
  print_i(p[0])  
}
```

// ポインタ

```
def main() {  
  var a:int = 0  
  goto b  
  print_i(2)
```

```
b:  
  print_i(1)  
}
```

```
def main() {  
  var a:int = 0  
  do {  
    a++  
    if (a < 5) continue;  
    print_i(a)  
  } while (a < 10);  
}
```

```
def main() {  
  var a:int = 0  
  do {  
    print_i(a)  
    a++  
    if (a > 10) break;  
  } while (true);  
}
```

```
def main() {  
  var a:int = 1  
  a is {  
    1? print_i(1) break  
    2? print_i(2) break  
  }  
}
```

# LLVMコンパイラ

## 意味解析入門

- [https://bytebucket.org/h\\_sakurai/ll/raw/bdd65a48683e3218222d67849f8661be9f421b6a/main.pdf](https://bytebucket.org/h_sakurai/ll/raw/bdd65a48683e3218222d67849f8661be9f421b6a/main.pdf)
- [https://bitbucket.org/h\\_sakurai/oll/raw/29a25147a7503cae6ef50c5b59c3dd3aa0893743/main.pdf?at=default](https://bitbucket.org/h_sakurai/oll/raw/29a25147a7503cae6ef50c5b59c3dd3aa0893743/main.pdf?at=default)
- 手順通りプログラムを書けばコンパイラが出来ると  
いうもの

# 出来事

- ・ Swiftがリリースされる
- ・ GCなくて動くATSはC言語を出力する関数型言語
- ・ GCのないRust1.0がリリース
- ・ CLRのネイティブコンパイラが発表される
- ・ Scalaのプログラマが少ないと言われている
- ・ LLVMを出力する文鳥言語と犬言語の開発者が現れる



# Camrup

- ・ OCamlは素晴らしいけど、構文が分かり辛い
- ・ AltJSが昨今はやっている
- ・ JSはアカデミックな理論が背景にない
- ・ アカデミックな背景のあるOCamlを出力するよくある言語を作ったらよいのでは？
- ・ <https://github.com/hsk/camrup/tree/master/docs/tutorial>



# Camrupの売り

- ・ Scalaに似たようなシンタックス
- ・ 部分的にはScalaを超えている `a(1)(2)(3)`は`a(1;2;3)`と書ける
- ・ モナド用構文はない
- ・ オブジェクト指向もあり、多相型も使える
- ・ コンパイルは高速
- ・ OCamlの型推論
- ・ ライブラリはOCamlの物が使える

# Camlupデモ

- [http://jsdo.it/h\\_sakurai/82IO](http://jsdo.it/h_sakurai/82IO)
- WebGLを使ったシューティングゲーム
- Camlup->OCaml->JavaScriptと変換して動いている

# ATSの研究

- ・ 線形論理でリソースを管理
- ・ 安全なプログラムを組み込みのプログラムに提供
- ・ 理論が知りたいので翻訳してみました
- ・ ATSのソースが汚いのでフォーマッタを調べてたのだけど、ML系のフォーマッタを作るのは難しい事が分かる



# Scalaコンパイラを作ろう

- ・ Scalaのコンパイラは遅いと言われています。
- ・ Haxeの作者はフリーの高速なActionScript2コンパイラMTASCをOCamlで作成して有名になりました。
- ・ HaxeもまたOCaml製で非常に高速です
- ・ ActionScript3のJava製コンパイラと比べても高速です。
- ・ OCamlでScalaのコンパイラを作れば高速になる可能性があります。

# Scalaコンパイラを作るには

- ・ JVMのバイトコードの読み書きが必要
- ・ JVM用のコンパイラの作成技術が必要
- ・ パーサの作成が必要
- ・ とりあえず基本的な事はやってみた

# Scalaパーサ

- ・ <https://github.com/hsk/scalaparser>
- ・ <https://github.com/hsk/scalaparser/blob/master/parser.mly>
- ・ 90%程は完了。コンフリクトが起きる箇所と、Unicode周りが問題
- ・ テストケース
- ・ <https://github.com/hsk/scalaparser/blob/master/main.ml>



# camljoos0

- ・ OCaml製のJavaサブセットのコンパイラ
- ・ <https://github.com/hsk/camljoos0>

