What is the best language for compiler writers

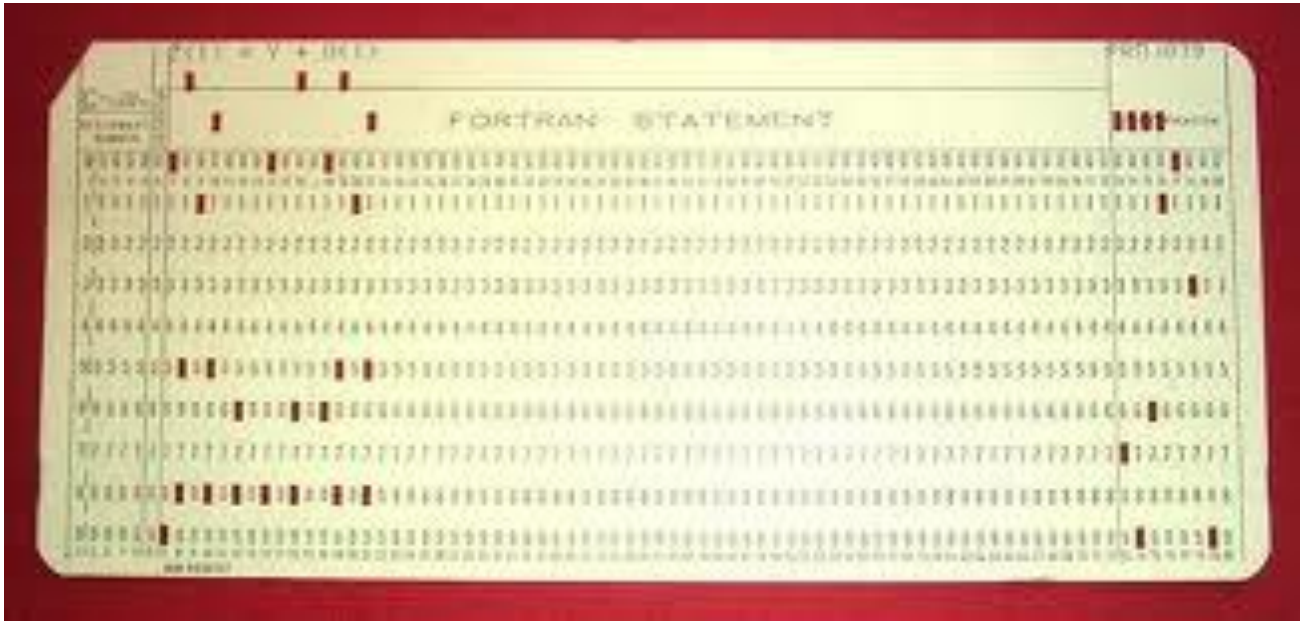
Seika Abe

MSI

The answer is Lisp !

Ratings (1984)

- Fortran ★
- Pascal ★★★★★



Why Pascal

1. Structured programming
2. Recursive call
3. User can define any data structures

SOLAR language compiler (1984 - 1985)

1. Implemented in C
2. byte-code compiler and virtual machine
3. Specially designed language used in a sewing company

Ratings (1985)

- Fortran ★
- Pascal ★★
- C ★★★★★
- Lisp ?

Why C

- Data structures can easily be defined
- Notation is very concise
- Simple and powerful macro processor
- Rich operators reflecting conventional hardware instructions
- Well designed pointer type system

Typing rules of C

$$\frac{e \Rightarrow (\tau\ *\cdot, \alpha, -)}{*e \Rightarrow (\tau, \mathtt{mem}(\tau, \alpha), \alpha)}\ (*)$$

$$\frac{e \Rightarrow (\tau, -, \alpha)}{\&e \Rightarrow (\tau\ *\cdot, \alpha, \mathtt{nil})}\ (\&)$$

$$\frac{e \Rightarrow (\tau\ *\cdot, \alpha, -) \qquad i \Rightarrow (\mathtt{int}\ \cdot, d, -)}{e \pm i \Rightarrow (\tau\ *\cdot, \alpha \pm d * \mathtt{sizeof}(\tau), \mathtt{nil})}\ (\text{ptadd})$$

$$\frac{e \Rightarrow (\tau\ *\cdot, \alpha, -) \qquad i \Rightarrow (\mathtt{int}\ \cdot, d, -)}{i + e \Rightarrow (\tau\ *\cdot, \alpha + d * \mathtt{sizeof}(\tau), \mathtt{nil})}\ (\text{ptadd}')$$

$$\frac{e_1 \Rightarrow (\tau\ *\cdot, \alpha_1, -) \qquad e_2 \Rightarrow (\tau\ *\cdot, \alpha_2, -)}{e_1 - e_2 \Rightarrow (\mathtt{int}\ \cdot, (\alpha_1 - \alpha_2)/\mathtt{sizeof}(\tau), \mathtt{nil})}\ (\text{ptdiff})$$

$$x[y] \triangleq *(x + y)$$

$$\frac{e \Rightarrow (\tau\ \cdot[n], -, \alpha)}{d_v(e) \Rightarrow (\tau\ *\cdot, \alpha, \mathtt{nil})}\ (\text{decay v})$$

$$\frac{e \Rightarrow (\tau\ \cdot(\cdots), -, \alpha)}{d_f(e) \Rightarrow (\tau\ (*\cdot)(\cdots), \alpha, \mathtt{nil})}\ (\text{decay f})$$

C quiz

With the follwing declaration,

```
int v[10], i,j,k;
```

Fill (?) to make the following two expressions
evaluate to the same value for any values of `i,j,k`.

```
v[i + j + k + 1]   and   (?)[i][j][k]
```

C compiler for Nintendo Entertainment System (NES) (ファミコン) (1986 - 1987)

- Impremented in C
- Native compiler for NES (6502 8-bits CPU)
- This project was not completed…

Ratings (1986)

- Fortran ★
- Pascal ★★
- C ★★★★
- Lisp ★?

Example of RTL in GCC

```
(insn/f 36 35 37 2 sum.c:4 (set (reg/f:SI 6 bp)
        (reg/f:SI 7 sp)) -1 (nil))

(insn/f 37 36 38 2 sum.c:4 (parallel [
            (set (reg/f:SI 7 sp)
                (plus:SI (reg/f:SI 7 sp)
                    (const_int -16
      [0xfffffff0])))
            (clobber (reg:CC 17 flags))
            (clobber (mem:BLK (scratch) [0 A8]))
        ]) -1 (nil))

(insn 5 2 6 2 sum.c:5 (set (mem/c/i:SI
       (plus:SI (reg/f:SI 6 bp)
                (const_int -4 [0xfffffffc]))
              [0 total+0 S4 A32])
        (const_int 0 [0x0])) 44 {*movsi_1}
   (nil))
```

Kyoto Common Lisp (KCl)

- Impremented in C
- by Masami Hagiya and Taichi Yuasa
- Conforms to CLTL1

Ratings (1987)

- Fortran ★
- Pascal ★★
- C ★★★
- Lisp ★★★★

Metal Slader Grory

Example of AGL program

```
begin
 if (save!=255) {
se_set(ANAUNSU); timer(1);
mprint ("[振り向き][EL__F][A__A]:
    :「このたびは　ATFこう空を;:
    :[振り向き][A__B]:
    :ごりよう　いただきまして;:
    :まことに　ありがとうございます;:
    :・・・当シャトルは　15．30分発;:
    :きどうターミナルステーション行きと;:
    :なっております・・・;:
    :りりくまで[EL__A]　もうしばらく　おまち下さい:
 :[振り向き]・・・」:
    :あずさ「[F__C]うき　うき・」[PAUSE][振り向き][A__A]"); }
command;
['みる']{
['あずさ']{ ser(pointer1,FIX,b1_13_a,b1_13_2a); exit; }
['エリナ']{ bun(b1_13_b); flag_mem1 = 1; exit; }
['まど']{ bun(b1_13_c); exit; }
pmenu; }
['はなす']{
['エリナ']{ bun (b1_13_d); flag_mem2=1; exit; }
['あずさ']{ bun (b1_13_e); flag_mem3=1; exit; }
pmenu; }
if(flag_mem1&&flag_mem2&&flag_mem3){
['さわる']{ ['エリナ']{ exit_menu; } pmenu; } }
```

AGL (Adventure Game Language) Compiler (1988 - 1990)

- Byte code compiler (in Lisp)
- Assembler (in Lisp)
- Incremental linker (in Lisp)
- Source level debugger (in Lisp)
- Scenario compressor (in C)
- Byte code executor (in assembler of NES)

Ratings (1990)

- Fortran ★
- Pascal ★★
- C ★★★
- Lisp ★★★★★ !

C compiler for Super Nintendo（スーパーファミコン）(1990 -1993)

1. Impremented in Lisp (KCl)
2. Native compiler for Super-NES 16 bits CPU (65816)
3. Retargetable compiler
    1. Target machines can easily be chaned
4. Graph coloring based register allocation
5. Basic code optimizations

A sad story

- The compiler was implemented in Lisp
- The retargetablility and portability was good
- It generated relatively good quality code

A sad story

- The compiler was implemented in Lisp
- The retargetablility and portability was good
- It generated relatively good quality code
- But, its compilation speed was very slow
- AND…

MOTHER 2

Translating to C **by hand**

1. Make light weight lisp kernel with GC as a C library
2. Using the library, translate Lisp code to C with performance care

Visit some sources of the compiler

Ratings (1994)

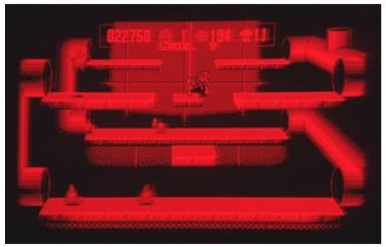
- Fortran ★
- Pascal ★★
- C ★★★
- Lisp ★★★★★ (! is removed)

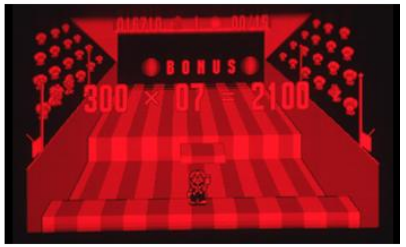
Today, I think the drawback of Lisp becomes not so serious problem,

by very huge improvements in hard wares
and
by very **small** improvements in Lisp compiler technologies.

C compiler for Virtual Boy (3D) (1994 - 1995)

- Implemented in C with the Lisp library
- V810 32bit CPU has a rich register set





COINS Project (2000 - 2005)

1. A big Compiler project in Japan
2. It aims at a mult-language, mult-target compiler
3. With some development tools and static program analizers
4. The system is implemented in Java

Java is a shit language

- Notation is very lengthy and ugly
- Constructing a data structure is very clumsy
- Sub-typing is not so useful
- Lack of disjoint union type (e.g. union in C)
- Name qualification depends its semantics
- No macro prosessor
- The name is the same to a toilet cleaner
- There’s more, but this margin is too narrow…

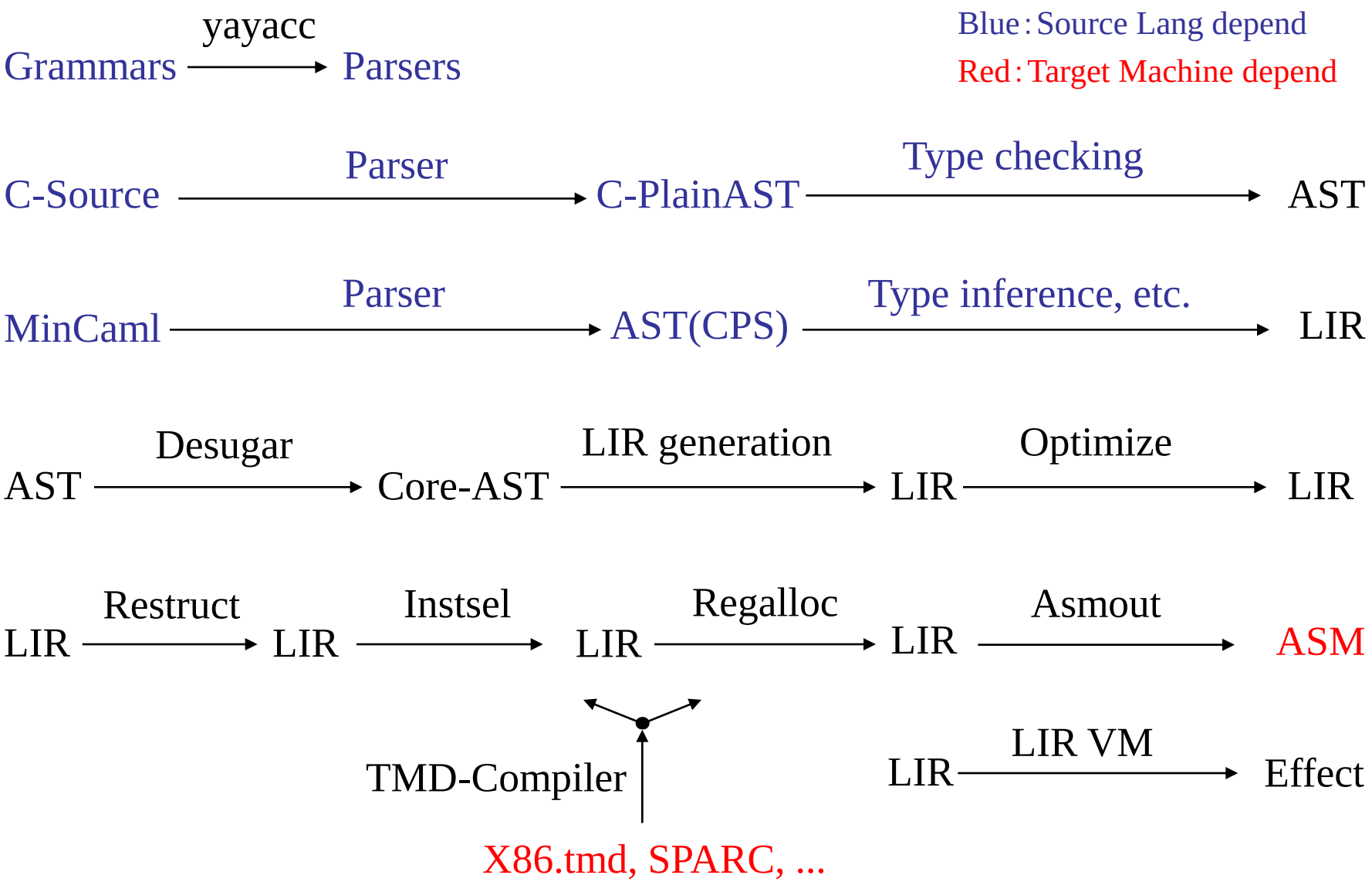
Ratings (2005)

- Fortran ★
- Pascal ★★
- C ★★★
- Lisp ★★★★★
- Java ★

SCK Project (2006 - 2007)

- A small compiler project of three members
- Sponsered by Mitou Project originated by Ikuo Takeuchi
- SCK stands for S-expression based Compiler Kit
- Impremented in Emacs Lisp (≒50000 lines)
- It aims to make Multi-sorce, Multi-target compiler, as COINS
- And to provide the compiler as a set of reusable translators connected by S-exp

SCK Compiler Kit



SCK demo

Portable Slice analizer (2012)

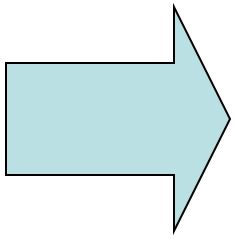
1. Impremented in Haskell
2. It is a library to analyze program slice

What is program slice

```
inw = FALSE;
nl = 0;
nw = 0;
nc = 0;
c = read();
while (c != EOF) {
  sticky
  nc++;
  if (c == '¥¥n')
    nl++;

  if (c==' '|c=='¥¥n'||c=='¥¥t')
    inw = FALSE;
  else if (inw == FALSE) {
    inw = TRUE;
    nw++; }

  c = read(); }
write(nl);
write(nw);
write(nc); <-SlicePoint
```



```
nc = 0;
c = read();
while (c != EOF) {
  sticky
  nc++;
  c = read(); }
write(nc);
```

Haskell may be a nice language for compiler writers

- It is a purely functional language!
- Normal order reduction
- Notation is very concise
- Strong typing
- Typs system is rich yet very easy to use

Ratings (2012)

- Fortran ★
- Pascal ★★
- C ★★★
- Lisp ★★★★★
- Java ★
- Haskell ★★?

Data flow analizer for Java (2012)

- Implemented (firstly) in Lisp
- It is a library to do global dataflow analysis of Java program

Translaring to Java **by hand**

- Make a small library to support writing lispy code in Java
- Using the library, translate Lisp code to Java

Thank you!