From Ideas to Practice

25 Feb 2021

(1) System F to Java

(2) type soundness to Java

(3) garbage collection to Java

(4) delimited continuations to Scala *

(5) intermediate rep. for imp. compilers

(6) program verification & semantics

(8) discussion

delimited continuations to scala

1974   run in BCPL/os

1984   ⎰ MF in "511"
1988   ⎱ APL "prompts"

1990   shift & reset

1992   N variants: Queinnec,
              Hieb & Dybvig, Filinski

2007   reconcile call/cc w/ "prompt"

2010   Scala *
_____

26-36 years

System F to Java generics

1974	System F; Λ
1984	structural subtyping; $\exists$
1985	Cardelli-Wegner
1989	F sub; recursion
1995	Pizza implemented
1998	GJ for compatibility
2001	Java release;
	2005 wide-spread use

27 years

type soundness in production

1978	the idea; basically PCF
1980	references; failed proof
1986	SML / NJ
1987	co-induction proof
1991/4	subject-red. proof
1995	Java, but casts

first PL design w/
explicit goal of soundness

17 years

automatic mem. mgt. (gc)

1963	Lisp at MIT
1973	Smalltalk
1970s	generational collection
1986	"conservative" gc
1991	Zorn: even conservative collection u faster than a good programmer
1995	Java w/ "conservative" gc
2000	Java w/ "precise" gc

32 - 37 years

Compiler representation (imperative)

	1960	flow charts (before)
	1973	data-flow analysis
	1980	data-flow graph
	1985	program-dependence graph (PDG)
····>	1990	sem. PDG vs SSA
		↑ ↑
		theory comp. algo.
	1995	FORTRAN res. compiler
◁··>	2005	LLVM

15 years;   33 years

Program Verification from Semantics

~1960s   VDM   ?

1968   Floyd Hoare : WP, SP

1972/6   LCF, Ng then, Typed Lisp

1981   verified Pascal compiler

1993   CL Inc. ACL2
verified HW/SW stack:
processor, "OS", compiler

2005   Comp Cert

2010   secure linux kernel

is it practical?

Discussion

① examples from your own
   readings

② is there a difference between
   user-facing innovations &
   internal ones?

③ Why decades?

④ PhD students:
        do you have patience?