Experiments in Formal IL Semantics

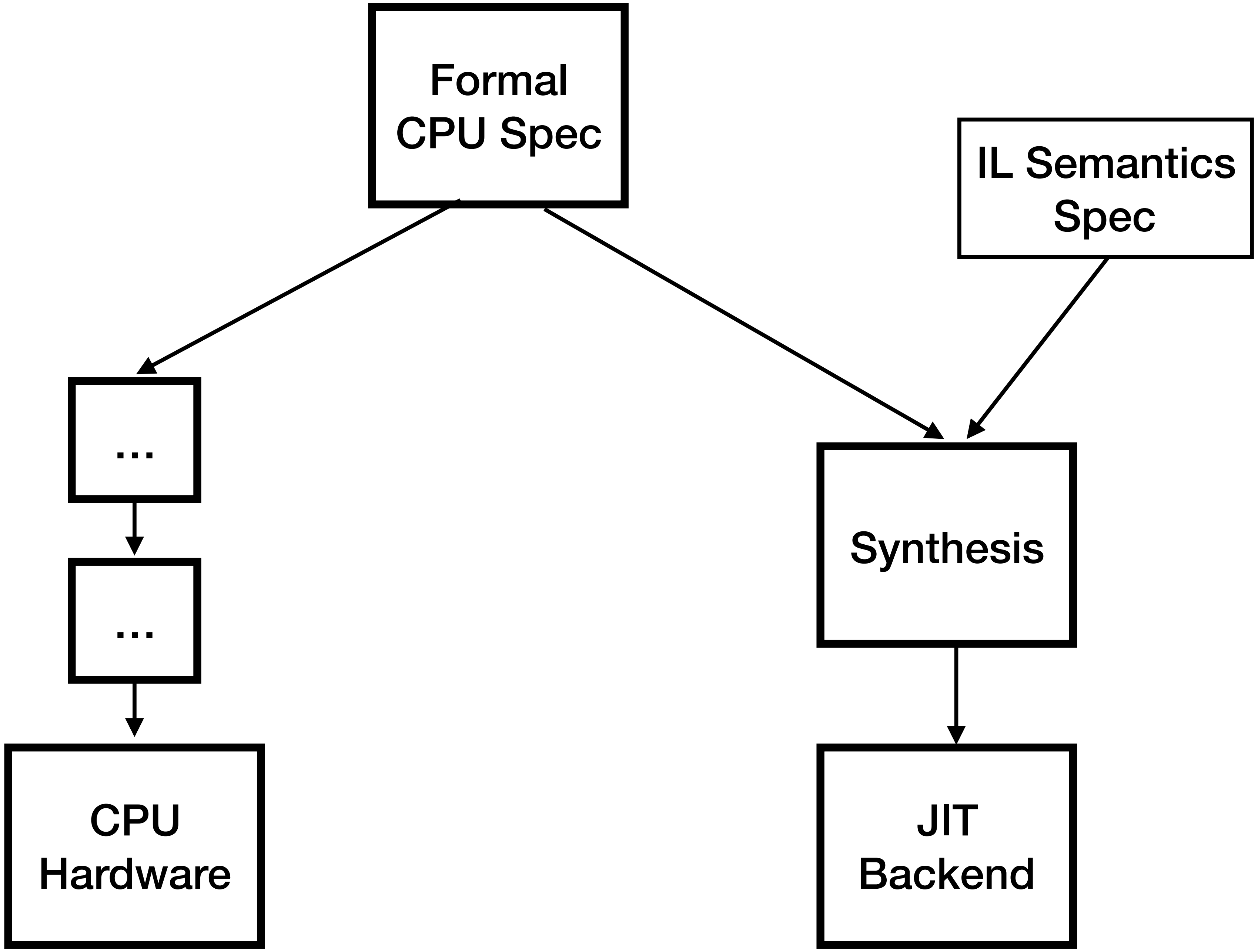
- NOT a lecture on how it “should” be done
- Author’s experiments in the last 6+ years

Approach 1

- “VLS: Programming a Smalltalk VM in Coq”
- Presented at IWST-2017 (Maribor, Slovenia)
- CompCert’s pass replacing MACH

Approach 2

- Target-Agnostic synthesis of backend



- First working PoCs for an academic Smalltalk (MT)
- 2014 — IWST, Cambridge, UK
- 2014 — Smalltalks, Córdoba, Argentina
- 2015 — IWST, Brescia, Italy
- 2016: enough to run ANSITester
- 2018: run Bee methods

Prior Art / Inspiration

- ArchC / AccGen: synthesizes binutils and LLVM backend from ISA spec
    - UC Berkeley research project
    - Custom DSL
    - Ad-hoc C++ solver
    - RTL Semantics

Prior Art / Inspiration

- Angr
    - Algebraic/symbolic execution of arbitrary binary code
    - Lifts CPU instructions to Valgrind VEX IR
    - Python on top of Z3 solver
    - Many ISAs
    - Possible superoptimizer approaches

Shingarov TA-VM

1. Parse ArchC’s PDL DSL

2. Assert facts into Prolog Database

3. PIG Solver:

    - Unify with IL (i.e. ST bytecodes) I/O-effects = op-semantics

    - Prolog + CLP($\mathbb{Z}$) for solving

    - Van Emden for re-writing [see next slide]

    - Uninterpreted symbols

- van Emden

```
:- op(500, xfx, =>).

e(X,Y) :- e2(X,Y).

e2(X,Z) :- e1(X,Y), e2(Y,Z).

e2(X,X).

/* Substitutive closure: */

e1(transfer(A1,B), transfer(A2,B)) :- e1(A1,A2).

e1(transfer(A,B1), transfer(A,B2)) :- e1(B1,B2).

/* Rewrite Axiom: */

e1(X,Y) :- (X => Y).

/* Instructions: */

<effect> => <instr> :- <conditions>.
```

- Uninterpreted symbols
    - Maximum freedom
    - Difficult to express complex arithmetic
    - E.g. rlwinm on PowerPC; contrast with VEX:

```
RLWINM r3,r1,0x1c,0x18,0x1f

t0 = GET:I32(gpr1)

t10 = shr32(t0,0x04)

t13 = shl32(t0,0x1c)

t9 = or32(t13,t10)

t7 = and32(t9,0xff)

PUT(gpr3) = t7
```

- Future: combine CLP($\mathbb{Z}$) with computer algebra (like *angr*)

Lessons

- Need a much simpler IL
- TR IL seems a good candidate

Discussion

- …