Harvard John A. Paulson
School of Engineering
and Applied Sciences

PL/HCI Seminar (252R/279R)

Programming Languages

whirlwind tour

Research in Programming Languages

- Coming with reusable insights and solutions.
- The problem is general and abstract. A problem space instead of a problem.
- Example: https://2019.splashcon.org/details/splash-2019-oopsla/2/Probabilistic-Verification-of-Fairness-Properties-via-Concentration

Venues

- POPL (principles of PL)
- PLDI (PL design and implementation)
- ICFP
- OOPSLA (SPLASH)
- ECOOP
- JFP, HOSC, TOPLAS
- workshops…

Mentoring Events

- PL mentoring workshop (PLMW): scholarship to attend conference + mentoring workshop.
- Oregon Programming Language Summer School (OPLSS).

What is programming?

- Manipulating Processes
  - Managing complexity
  - Controlling outcomes
- Thoughts as computation
  - Reasoning
  - Procedural vs. declarative knowledge

A language has

- primitives,
- means of combination,
- means of abstraction.

Back in time…

- *The acts of the mind, wherein it exerts its power over simple ideas, are chiefly these three: 1. Combining several simple ideas into one compound one, and thus all complex ideas are made. 2. The second is bringing two ideas, whether simple or complex, together, and setting them by one another so as to take a view of them at once, without uniting them into one, by which it gets all its ideas of relations. 3. The third is separating them from all other ideas that accompany them in their real existence: this is called abstraction, and thus all its general ideas are made.*

- John Locke, An Essay Concerning Human Understanding (1690)

ex: arithmetic

- primitives: 0, 1, 2, 3, …
- means of combination: +, *, −, …
- means of abstraction: x, y, z, …

ex: first-class functions

- primitives: 0, 1, 2, 3, …, +, *, -
- means of combination: function application, ()
- means of abstraction: function abstraction, define

Pan

- http://conal.net/papers/functional-images/
- primitives: an image is a function from coordinates to pixel.
- means of combination: composing images is composing functions.
- means of abstraction: as defined by host language.

Compositional Meaning

- Syntax
- Meaning is syntax-driven, and take meaning of parts to meaning of whole.
- Denotational semantics

ex: arithmetic

- [syntax] = number
- primitives: 0, 1, 2, 3, …
  [n] = n
- means of combination: +, *, −, …
  [t1 + t2] = [t1] + [t2]
- means of abstraction: x, y, z, …
  [x] = lookup x in env???

ex: arithmetic

- [syntax] = env => number
- primitives: 0, 1, 2, 3, …
  [n] = e => n
- means of combination: +, *, −, …
  [t1 + t2] = e => [t1]e + [t2]e
- means of abstraction: x, y, z, …
  [x] = e => lookup x in e

Semantics

- static semantics (e.g. type system)
- dynamic semantic (e.g. small-step or big-step)

Formal Reasoning

- Coq, Agda, …, pencil & paper
- https://softwarefoundations.cis.upenn.edu/plf-current/toc.html
- Curry–Howard Isomorphism

Verification

- Frama-C
- Dafny
  https://rise4fun.com/dafny
- F*
  https://www.fstar-lang.org/tutorial/

Synthesis

- Survey: https://rishabhmit.bitbucket.io/papers/program_synthesis_now.pdf
- Some applications:
  - data wrangling: FlashFill https://microsoft.github.io/prose/
  - graphics: SKETCH-N-SKETCH https://ravichugh.github.io/sketch-n-sketch/releases/icfp-2018-svg/

Logic Programming

- miniKanren
  - http://io.livecode.ch/learn/webyrd/webmk
  - http://io.livecode.ch/learn/gregr/icfp2017-artifact-auas7pp
- Prolog
- Constraint programming

Probabilistic Programming

- webPPL
  - http://webppl.org/
  - http://dippl.org/
- Hansei
  - http://okmij.org/ftp/kakuritu/Hansei.html
- Hakaru
  - https://hakaru-dev.github.io/

What are the PLs in HCIs?

- primitives, means of combination, means of abstraction
  - vs. just primitives (think gestures, clicks, etc.) that do not compose well
- semantics? reasoning? verification? synthesis? constraints? probabilities?
- communicating with a computer is communicating with a process that “blindly” manipulates symbols.