Semantics of Flow Analysis

What does a flow analysis mean? ← More precisely: a locally consistent set of annotations.

Result of flow analysis is a proposition — To optimize e to e', want to show $P \supset \mathcal{E}[\![e]\!] = \mathcal{E}[\![e']\!]$ ($e \simeq e'$ op eqn)

What proposition does a flow analysis represent?
How to distinguish correct from incorrect data flow equations?

Use propositions as types.
- Get soundness from results in type theory (free!)
- Now can justify optimization
- Don't get reconstruction/analysis algorithm

vs. Abstract interp
- harder
- not always done
- this comes for free

- Read abs interp fns as conjunctive types: $f \in (\forall x : \hat{D}).(\hat{f}(x))$ eg $*: pos \to pos \to pos$, $*: pos \to neg \to neg$, etc
- Local consistency minimizes type rules

Examples

Strictness Analysis	$\Delta, \{\perp\}$	or dmony types (Know-Mishra)
Dead Code/Useless Var Anal	$\Delta, \cup$	PER types
Ref Ct Anal	$\{\inf, sn\}$ of types	Harrier PER types. - Correctness wrt "instrumented semantics",
Avail Exp, Red in Strength	$x = y+z$	invariants, not just types — not std semantics
Binding Time Analysis		2-level types — Justification for optimizations?

Open Questions

- Higher Order / Closure Analysis — this should be clearer than Abs Interp case (?)
- Data Structures / Abstract Locations

- Does this break down on the next example?
- Does it help?