What is program analyzer? Programs that reason about programs.

Requirements Soundness (to prevent FN), Completeness (to prevent FP), Termination

Rice's theorem Non-trivial property is undecidable => Approximation

1

2 **Type Analysis** AST => Type **constraints** => unifier (union-find) flow-insensitive

Others: CFG => **constraints** => solver flow-sensitive

Lattice and Fixed-points 3

Partial order, unique ⊥ and ⊤,
x⊔y and x⊓y exist for all x,y ∈S

Kleene's fixed-point theorem:
f has a unique least fixed-point if
$f: L \rightarrow L$ is monotone, L is a lattice with finite height

Each iteration of *n*-node CFG: $f: L^n \rightarrow L^n$ worklist algorithm to solve fixed point

4 **Flow Sensitive Analysis** Forward (available exp, R-D)/backward(liveness, very busy exp), lattice, constraint rules for nodes, join operator, may(R-D, liveness)/must (available exp, very busy exp)

Widening and Narrowing 5
(Interval analysis)

Widening: infinite height of lattice → finite
Narrowing: improve accuracy

Control Flow A. 8
Closure analysis
Cubic framework
Func.Pointer
OOP

6 **Path Sensitive**
assert: conditional constraints
relational analysis: relational constraints

Interprocedural Analysis 7
Interprocedural CFGs: exist invalid paths
context-sensitive IPA: context(callchain, data)

What is program analyzer? Programs that reason about programs.

Requirements Soundness (to prevent FN), Completeness (to prevent FP), Termination

Rice's theorem Non-trivial property is undecidable => Approximation

1

2 **Type Analysis** AST => Type **constraints** => unifier (union-find) flow-insensitive

Others: CFG => **constraints** => solver flow-sensitive

Lattice and Fixed-points 3

Partial order, unique ⊥ and ⊤,
x⊔y and x⊓y exist for all x,y ∈S

Kleene's fixed-point theorem:
f has a unique least fixed-point if
$f: L \rightarrow L$ is monotone, L is a lattice with finite height

4 **Flow Sensitive Analysis** **Widening and Narrowing** 5

6 **Path Sensitive** **Interprocedural Analysis** 7 **Control Flow A.** 8

9 **Pointer Analysis**
Andersen's: direction (=)
Steensgaards's: undirection (=)
Interprocedural PA, Null pointer
Flow-sensitive PA, Escape Analysis

10 **Abstract Interpretation vs. Symbolic Exe.**
Symbolic value
Concrete execution → Abstract execution