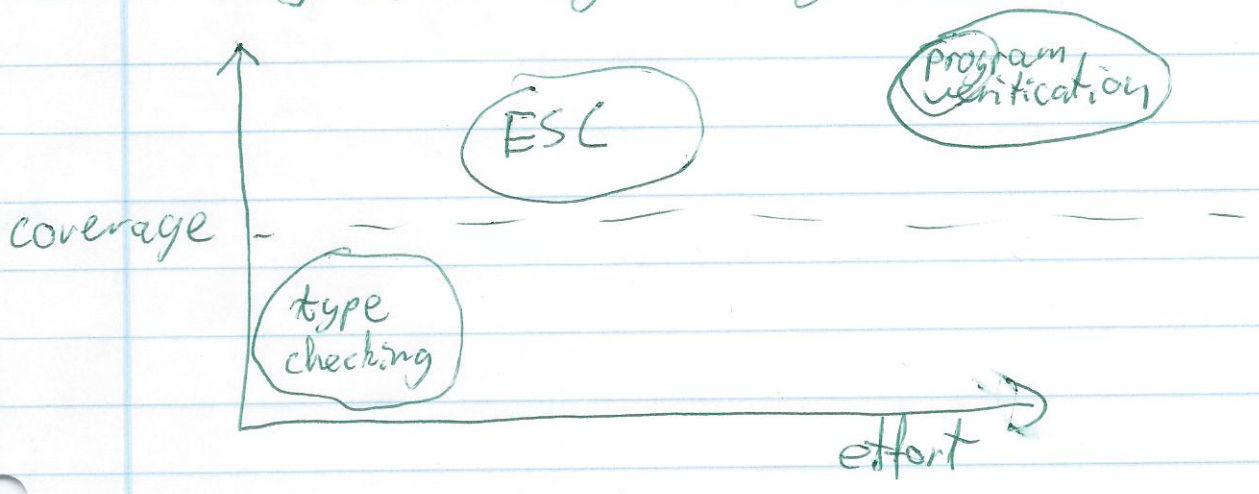
KLADÉ

ESC / Java2 - Extended static checking for java

Motivation

- Pragmatic bug finding for Java



How it works

Pre conditions
Post
Invariants
} JML

/*@ non_null */ Object O

annotated java program

Front end
↓ AST
Transformer
↓ GC ≈ TiP = desugaring
VC Generator
↓ VC
Theorem Prover
↓ results
Post Processor
↓ output

Type specific Background Predicate

universal background Predicate

Weakest Precondition

$P_m \Rightarrow P = WP(m, Q)$

Pre invariants

Post invariants

- $WP(x=E, Q) = Q[x|E]$

Blow up

$WP(if(E)\{S_1\}else\{S_2\}, Q) =$

$E \Rightarrow WP(S_1, Q) \wedge \neg E \Rightarrow WP(S_2, Q)$

↑ fixed by ESC somehow

Proving

$UBP \wedge BP_T \Rightarrow VC$

Classification

Unsound / Uncomplete
- no overflow
- aliasing
- 1 ½ loop unroll
- 1.0 ≠ 2.0 cannot be proved

Expresiveness
Safety properties
assert

Modularity
Intra procedural
Scales well - larger methods are harder

Lightweight / heavyweight
No annotations
Feels lightweight - is heavyweight