UFRR

UNIVERSITY OF OXFORD

UNIVERSITEIT STELLENBOSCH UNIVERSITY

UFAM

DepthK: A k-Induction Verifier Based on Invariant Inference for C Programs (Competition Contribution)

William Rocha, **Herbert Rocha**, Hussama Ismail, Lucas Cordeiro, and Bernd Fischer

ETAPS
EUROPEAN JOINT CONFERENCES ON THEORY & PRACTICE OF SOFTWARE

TACAS 2017

Competition on Software Verification (SV-COMP)

DepthK: K-Induction + Invariant Inference

DepthK employs **Bounded Model Checking** (BMC) and ***k*-Induction** based on program invariants, which are automatically generated using **polyhedral constraints**

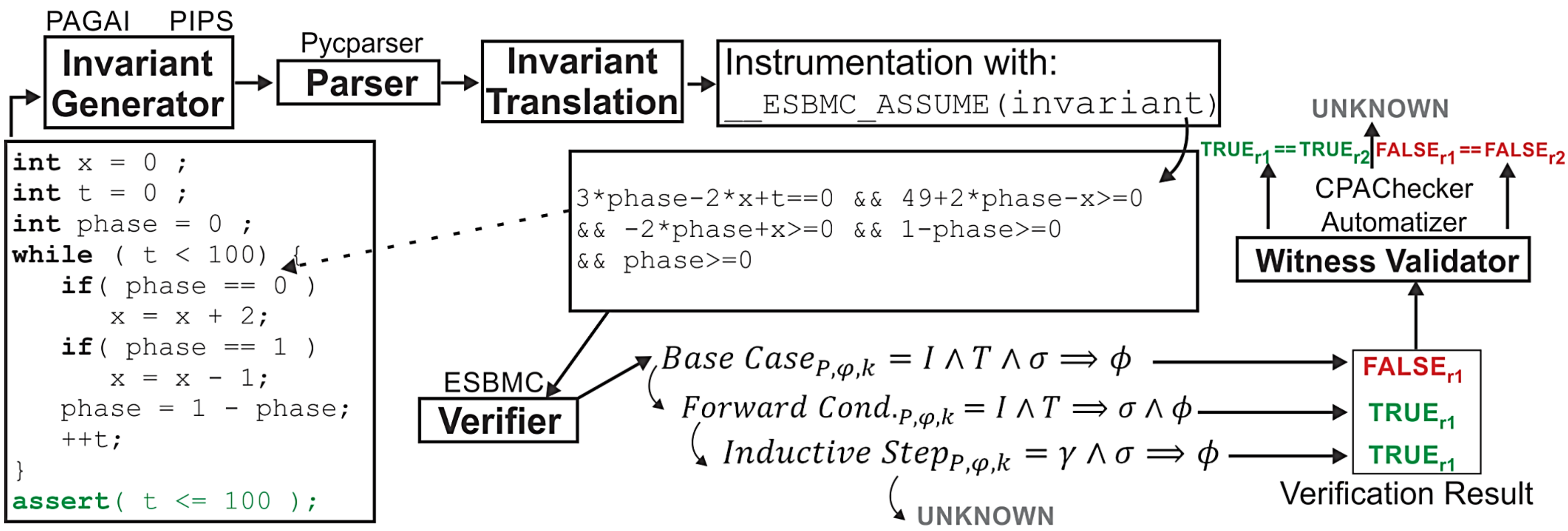
- ✓ DepthK uses ESBMC, a context-bounded symbolic model checker that verifies single- and multi-threaded C programs
- ✓ The *k*-induction step: base case, forward condition and inductive step
- ✓ DepthK uses PAGAI (SVCOMP'17) and PIPS tools to infer program invariants
- ✓ DepthK integrates the witness checkers CPAchecker and Ultimate Automizer for checking verification results

Verification Approach

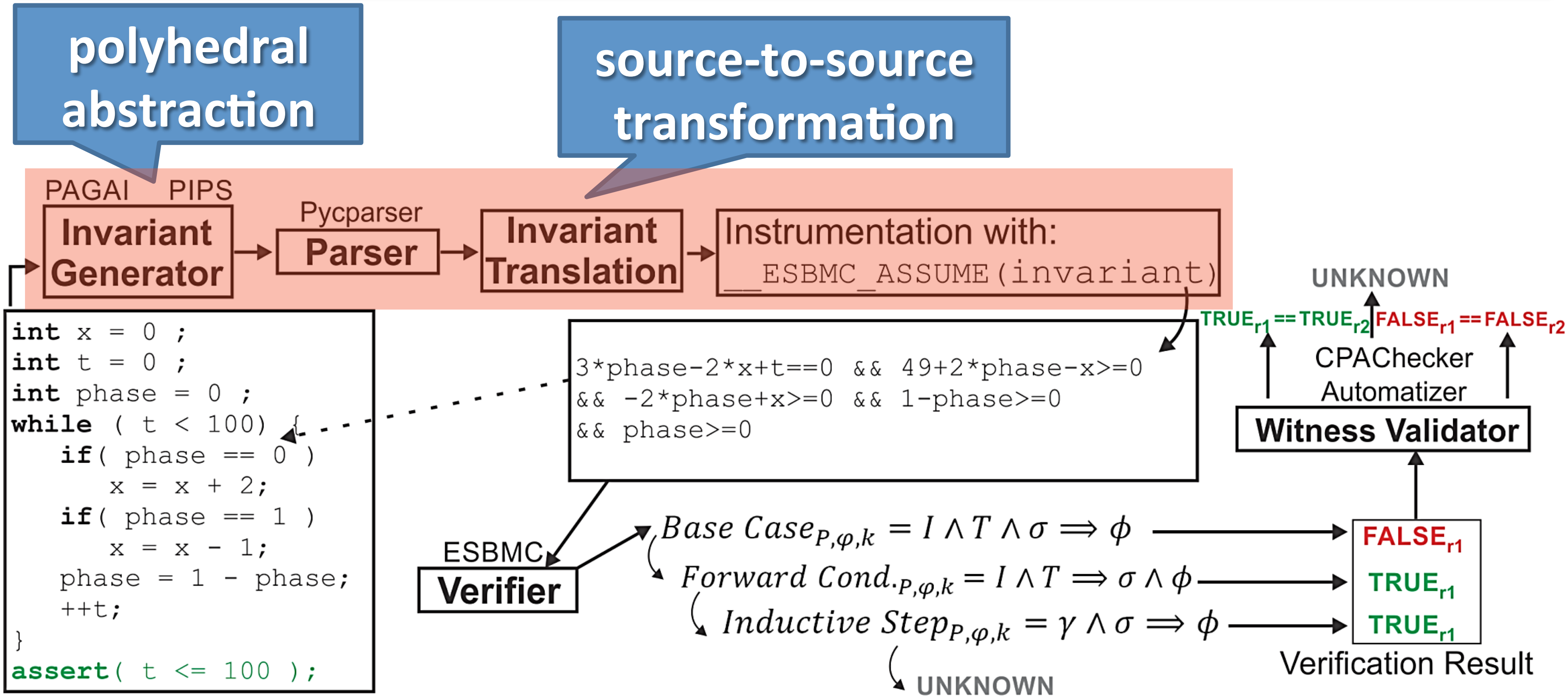
DepthK is a **source-to-source transformation** tool that extends ESBMC to falsify or prove correctness of a given (safety) property for any depth without manual annotation of **loop invariants**

- ✓ PAGAI applies source code analysis to infer invariants for each control-flow point of a C program using LLVM
- ✓ In PIPS, for each program instruction, the polyhedral invariants are propagated along with instructions, using the previously computed transformers

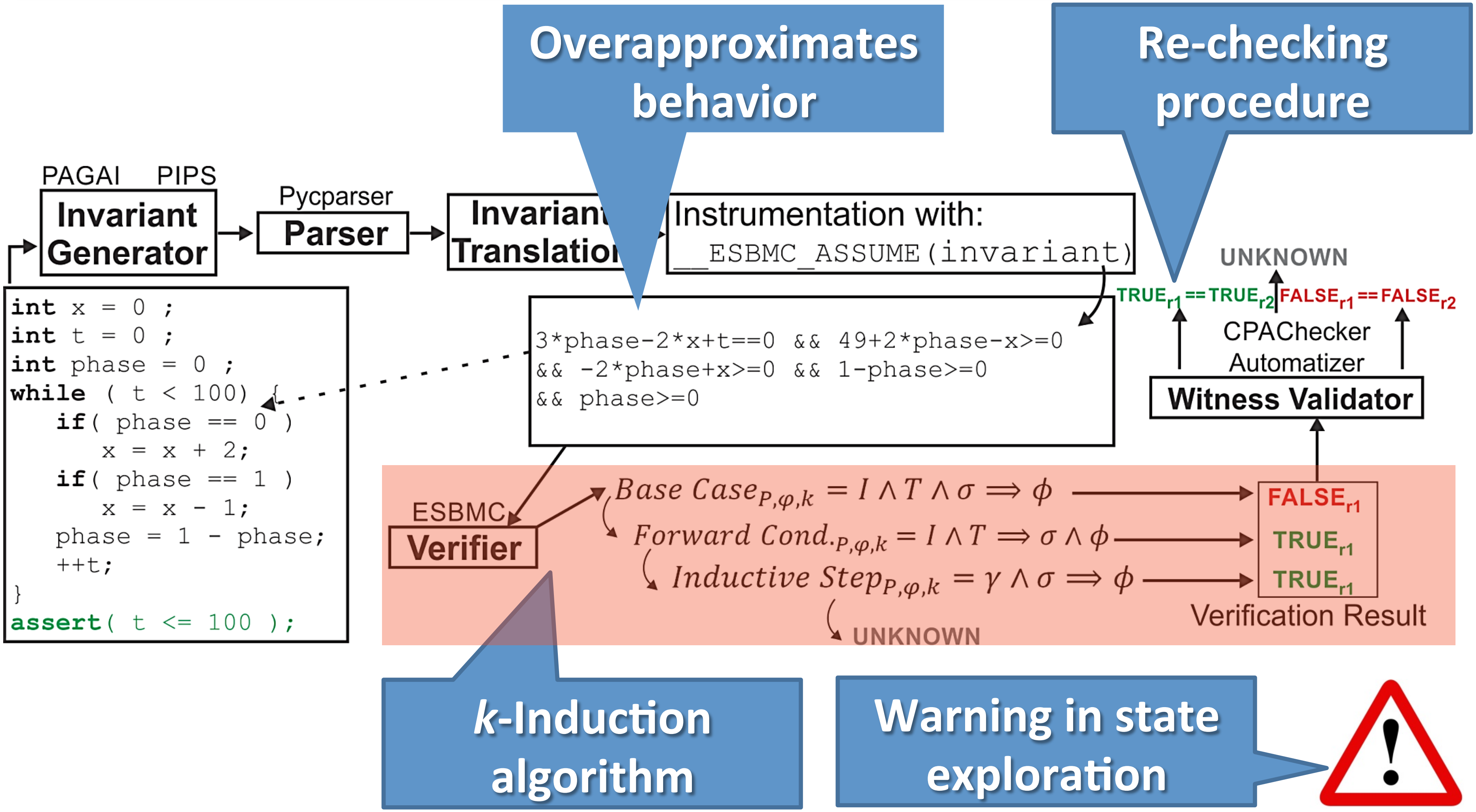
Verification Approach



Verification Approach



Verification Approach



Strengths and Weaknesses

- ✓ The **tool lies in the combination** of the ***k*-induction algorithm** with **program invariants** to specify pre- and post-conditions

- ✓ In preliminary experiments, **PAGAI/PIPS** tools were unable to produce **inductive invariants** for the *k*-induction algorithm, either due to a **weak transformer** or **not convex invariants**

- ✓ All incorrect answers produced by our tool in the competition are due to **bugs in its implementation**
  - Witness validation issues to confirm DepthK results
  - Trace back the data in the source code transformation

Strengths and Weaknesses

SV-COMP'17 results:

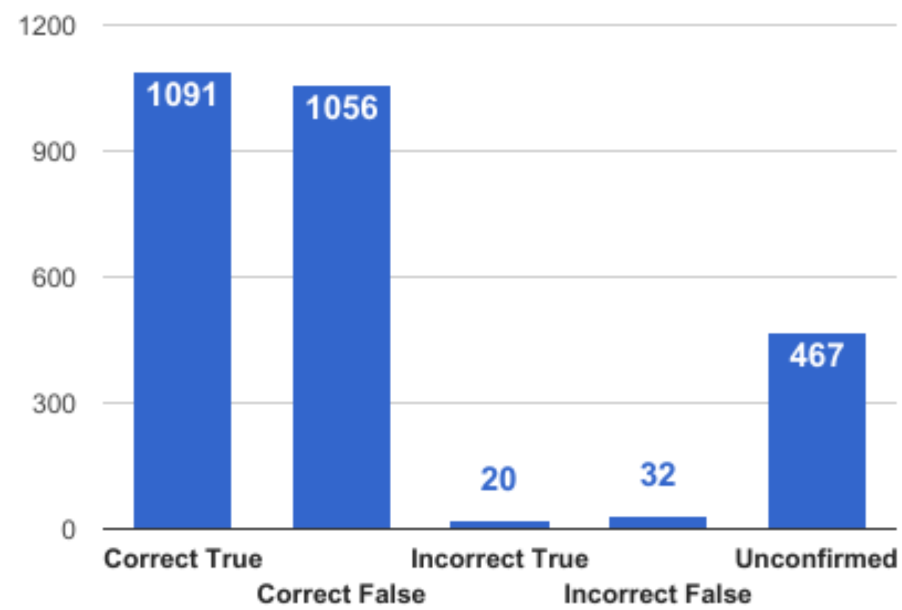
✓ Improvements over "**plain**" ESBMC

✓ DepthK outperforms all **ESBMC** versions in:

- ReachSafety-BitVectors
- ReachSafety-Heap
- ReachSafety-Loops
- MemSafety-Arrays

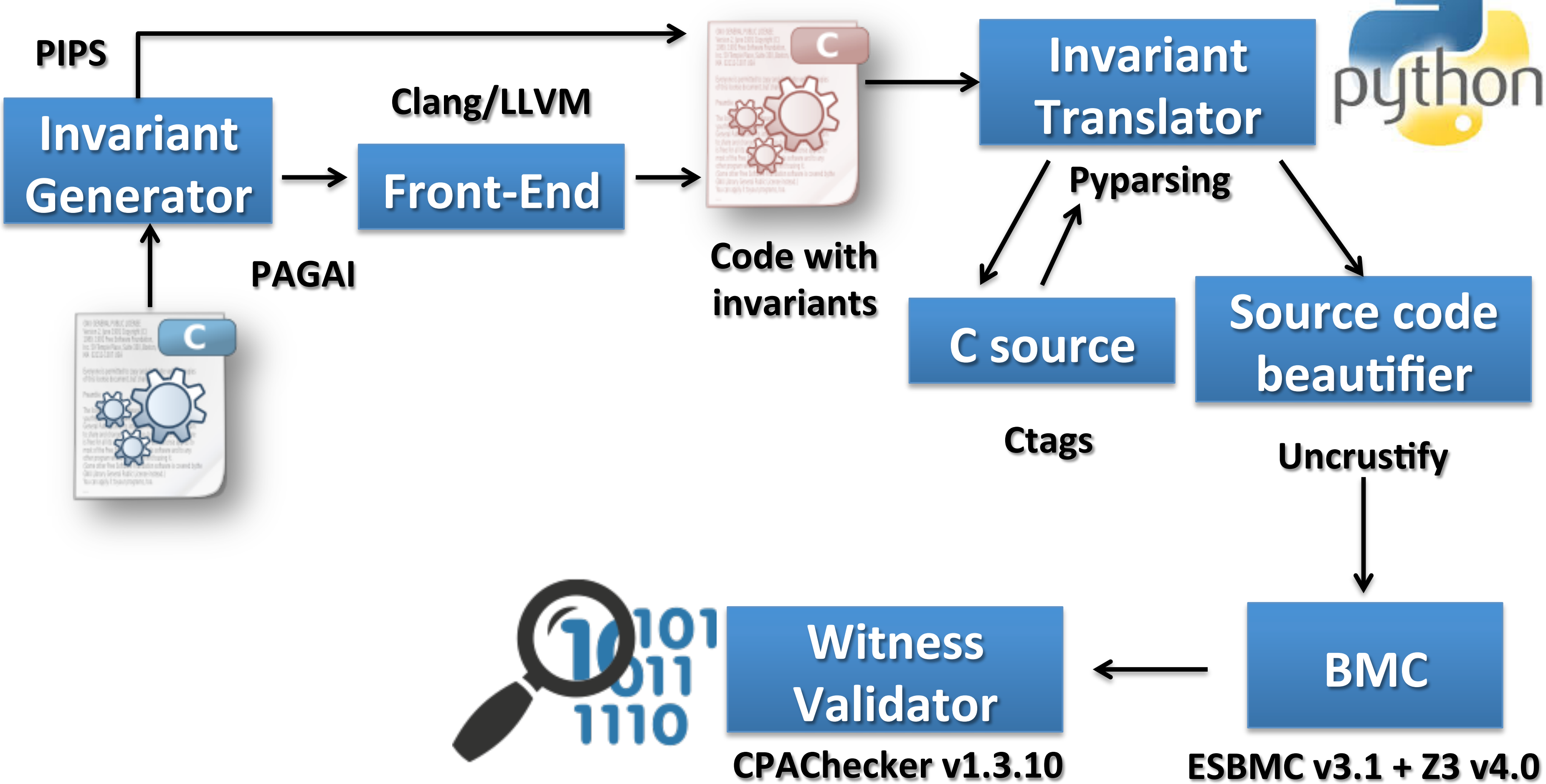
✓ DepthK outperforms **CPA-kInd**:

- ReachSafety-Heap
- ReachSafety-Recursive
- Overflows-BitVectors
- Category **FalsificationOverall**

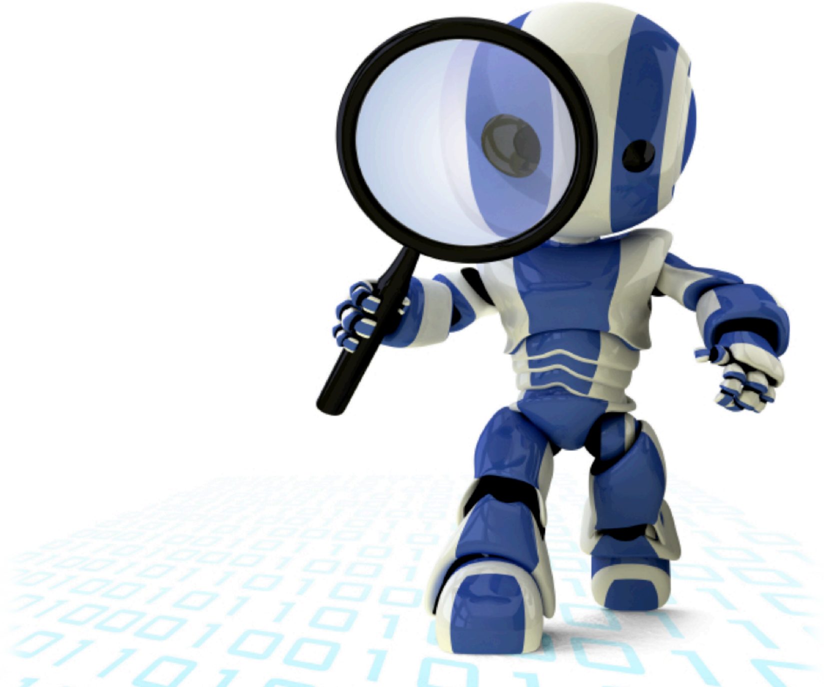


Architecture, Implementation and Availability

Source-to-Source Transformation



DepthK tool is available at **https://github.com/hbgit/depthk/archive/depthk v3.tar.gz**



Thank you for your attention!

herberthb12@gmail.com

https://github.com/hbgit/depthk