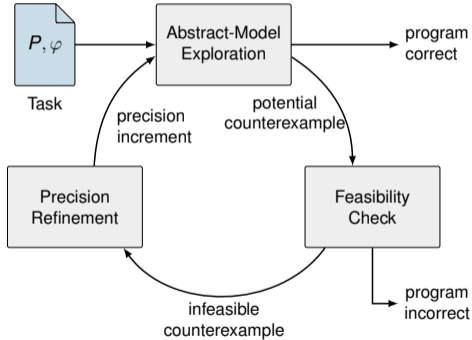


Component-based CEGAR

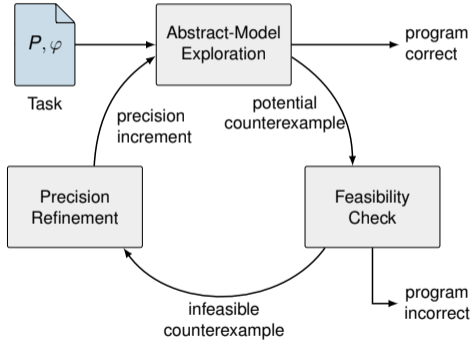
(under review)

Dirk Beyer, Jan Haltermann,
Thomas Lemberger, Heike Wehrheim
01.10.2021

Motivation: Classic CEGAR



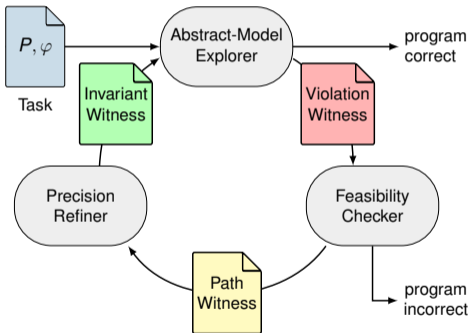
Motivation: Classic CEGAR - Problem



Problem:

- Many tools employ CEGAR (statefull)
- Common underlying schema
- New idea \Rightarrow New Implementation

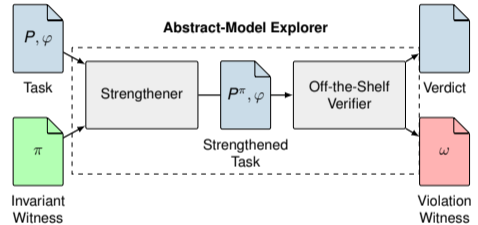
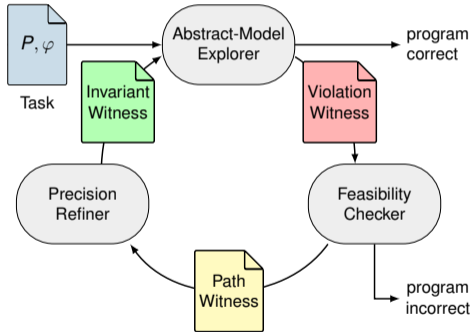
Component-based CEGAR (C-CEAR)



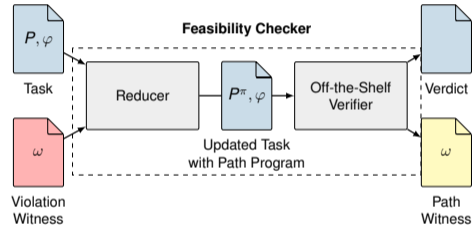
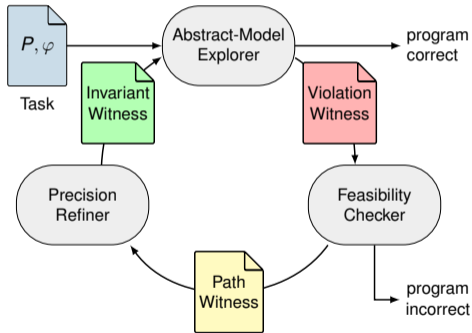
Defined components and interfaces:

- Components (stateless):
 - Abstract Model Explorer
 - Feasibility Checker
 - Precision Refiner
- Interfaces (existing formats):
 - Violation Witness
 - Path Witness
 - Invariant Witness
- Construction for off-the-shelf components

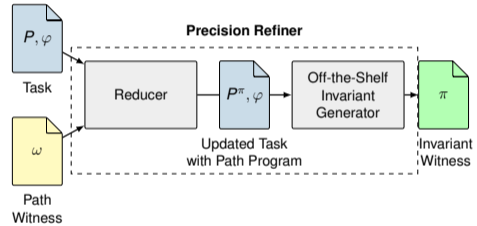
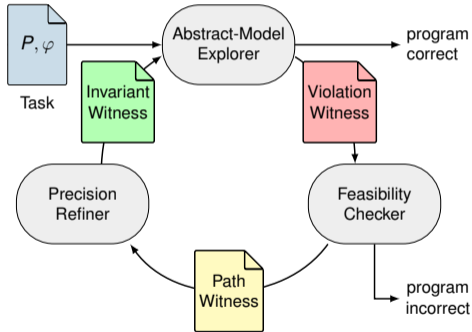
Usage of Off-the-Shelf Components - Model Explorer



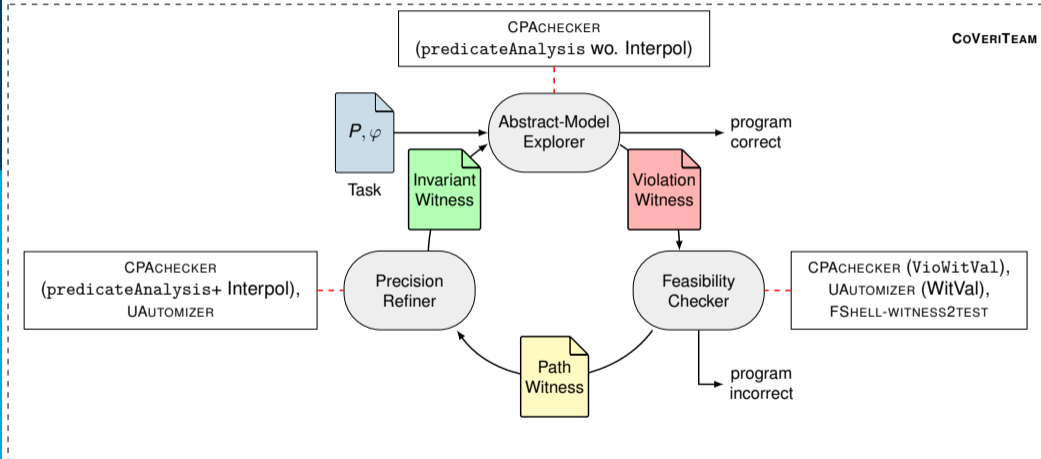
Usage of Off-the-Shelf Components - Feasibility Checker



Usage of Off-the-Shelf Components - Precision Refiner



Implementation



Evaluation

Research Questions:

- RQ1: Overhead of a component-based approach (with predmap)
- RQ2: Cost using existing formats
- RQ3: Use of off-the-shelf components

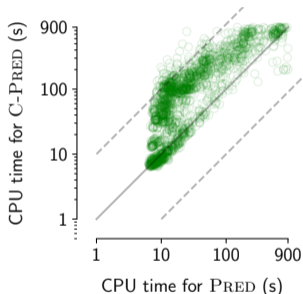
Dataset: SV-BENCHMARKS (4510 tasks), SV-COMP setting

RQ1: Overhead of component based design

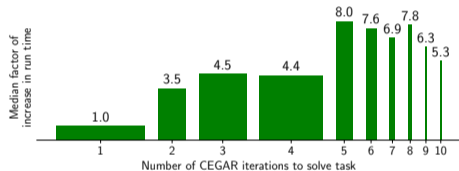
	overall	correct proof	alarm	incorrect proof	alarm
Pred	2 183	1 343	840	0	7
C-Pred	2 105	1 297	808	0	2

(Using predmap as exchange format)

RQ1: Overhead of component based design



Comparison of CPU time for
Pred and C-Pred

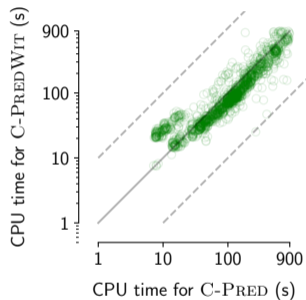


Median factor of run-time increase by
C-Pred compared to Pred.
Overall median increase is 3.2

RQ2: Cost of Standardized Formats

	overall	correct proof	alarm
C-Pred	2 105	1 297	808
C-PredWit	1 573	978	595

- Effectiveness reduces by 25%
- Reasons:
 - Not all predicates discovered are exported
 - No loop unrollings in witness



Comparison of CPU time for
C-Pred and C-PredWit

RQ3: C-Cegar using different components

RQ 3.1: C-PredWit + different feasibility checker (with precision refiner CPACHECKER)

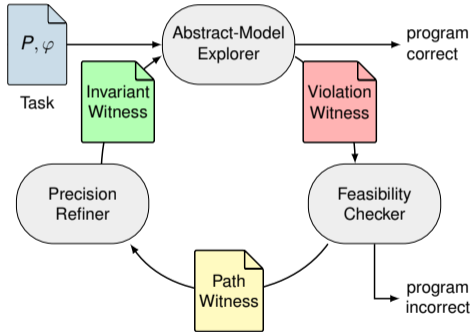
	overall	correct			unique
		proof	unique	alarm	
CPACHECKER	1 573	978	61	595	317
FSHELL-WITNESS2TEST	612	515	0	97	56
UAUTOMIZER	1 225	918	1	307	20

RQ3: C-Cegar using different components

RQ 3.2: C-PredWit + different precision refiner (with feasibility checker CPACHECKER)

	overall	proof	unique	correct alarm	unique
CPACHECKER	1 573	978	301	595	303
UAUTOMIZER	1 016	716	41	300	6

Summary - C-CEGAR



- Clear defined components and interfaces
- Implementation in COVERTEAM
- Evaluation show advantages:
 - Same expressiveness (with lower efficiency (3.2))
 - Existing formats can be used
 - Usage of existing off-the-shelf components
- Starting point for further development