Databases and Software Engineering

Hochschule Harz
Harz University of Applied Sciences

MENSCH | TECHNIK
ORGANISATION | PLANUNG

PCLocator: A Tool Suite to Automatically Identify Configurations for Code Locations

Elias Kuiter, Sebastian Krieter, Jacob Krüger, Kai Ludwig, Thomas Leich, Gunter Saake

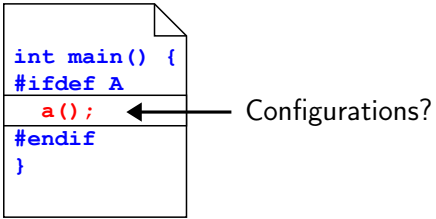
University of Magdeburg, Harz University of Applied Sciences, METOP GmbH

SPLC 2018
September 10–14 | Gothenburg, Sweden

Challenge

The Challenge

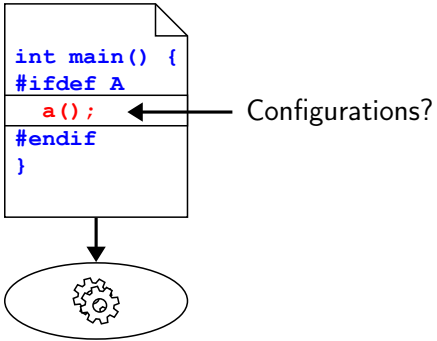
Given a specific **program location** in the source code,



Challenge

The Challenge

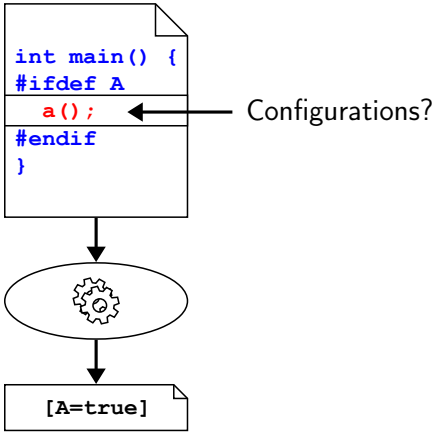
Given a specific **program location** in the source code, can you apply automatic analysis techniques



Challenge

The Challenge

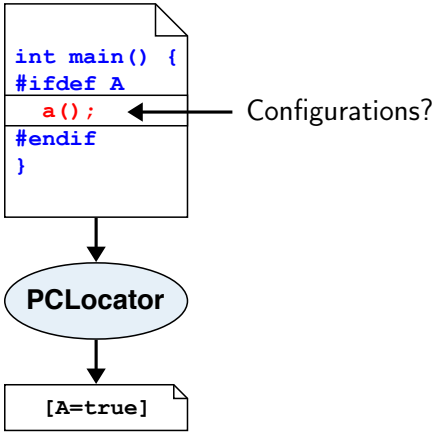
Given a specific **program location** in the source code, can you apply automatic analysis techniques to find **concrete configurations** that include the program location in question?



Challenge

Presence Condition Locator Tool Suite

- Static analysis of C code
- Uses existing tools
- Analyzes preprocessor and build system usage
- Open source, simple usage
- freely available on GitHub



Background

Under which conditions is a program location included?

Source Code Annotations

- For fine-grained variability
- C preprocessor directives (`#ifdef`)
- Results in propositional formulas (*presence conditions*)

Background

Under which conditions is a program location included?

Source Code Annotations

- For fine-grained variability
- C preprocessor directives (`#ifdef`)
- Results in propositional formulas (*presence conditions*)

```
/* Autodetects gzip/bzip2 formats. fd may
     be in the middle of the file! */
#if ENABLE_FEATURE_SEAMLESS_LZMA \
 || ENABLE_FEATURE_SEAMLESS_BZ2 \
 || ENABLE_FEATURE_SEAMLESS_GZ \
 /* || ENABLE_FEATURE_SEAMLESS_Z */
    extern void setup_unzip_on_fd(int fd);
#else
#define setup_unzip_on_fd(...) ((void)0)
#endif
```

Background

Under which conditions is a program location included?

Source Code Annotations

- For fine-grained variability
- C preprocessor directives (`#ifdef`)
- Results in propositional formulas (*presence conditions*)

```
/* Autodetects gzip/bzip2 formats. fd may
     be in the middle of the file! */
#if ENABLE_FEATURE_SEAMLESS_LZMA \
 || ENABLE_FEATURE_SEAMLESS_BZ2 \
 || ENABLE_FEATURE_SEAMLESS_GZ \
 /* || ENABLE_FEATURE_SEAMLESS_Z */
    extern void setup_unzip_on_fd(int fd);
#else
#define setup_unzip_on_fd(...) ((void)0)
#endif
```

Background

Under which conditions is a program location included?

Build System

- For coarse-grained variability
- Specified in *Makefiles*
- Results in *presence conditions* for files

Background

Under which conditions is a program location included?

Build System

- For coarse-grained variability
- Specified in *Makefiles*
- Results in *presence conditions* for files

```
lib-y:=
lib-$(CONFIG_ARP)         += arp.o interface.o
lib-$(CONFIG_ARPING)      += arping.o
lib-$(CONFIG_BRCTL)       += brctl.o
lib-$(CONFIG_DNSD)        += dnsd.o
lib-$(CONFIG_ETHER_WAKE)  += ether-wake.o
lib-$(CONFIG_FAKEIDENTD)  += isrv_identd.o isrv.o
lib-$(CONFIG_FTPD)        += ftpd.o
lib-$(CONFIG_FTPGET)      += ftpgetput.o
lib-$(CONFIG_FTPPUT)      += ftpgetput.o
```

Background

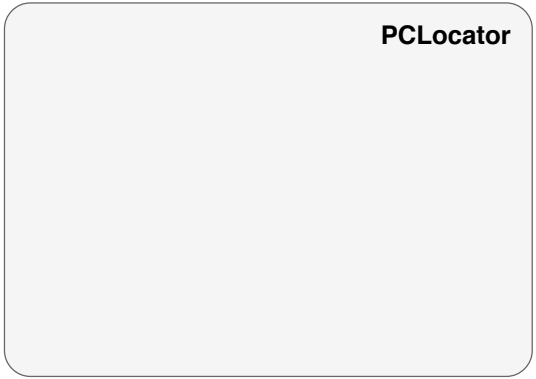
Under which conditions is a program location included?

Build System

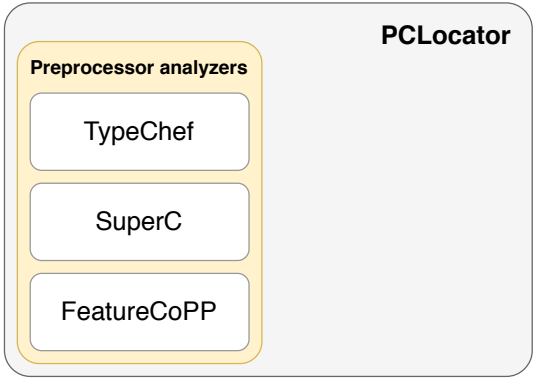
- For coarse-grained variability
- Specified in *Makefiles*
- Results in *presence conditions* for files

```
lib-y:=
lib-$(CONFIG_ARP)          += arp.o interface.o
lib-$(CONFIG_ARPING)       += arping.o
lib-$(CONFIG_BRCTL)        += brctl.o
lib-$(CONFIG_DNSD)         += dnsd.o
lib-$(CONFIG_ETHER_WAKE)   += ether-wake.o
lib-$(CONFIG_FAKEIDENTD)   += isrv_identd.o isrv.o
lib-$(CONFIG_FTPD)         += ftpd.o
lib-$(CONFIG_FTPGET)       += ftpgetput.o
lib-$(CONFIG_FTPPUT)       += ftpgetput.o
```

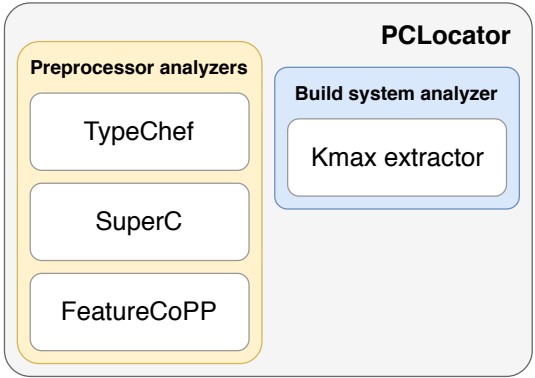
Architecture



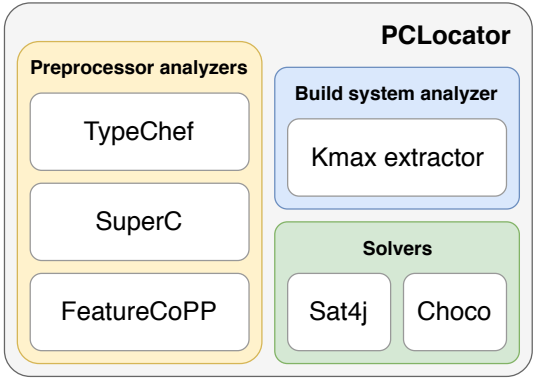
Architecture



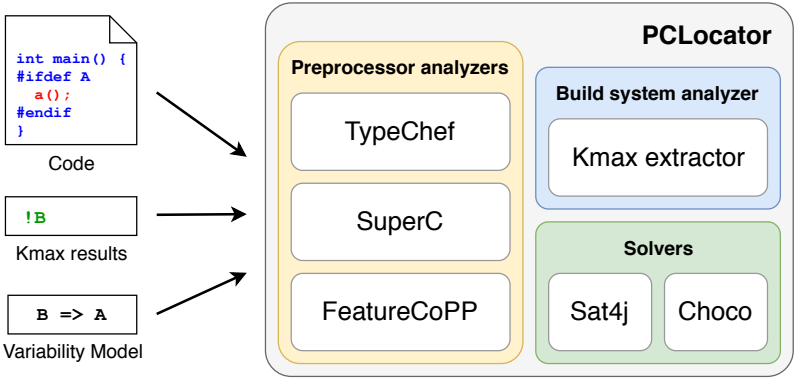
Architecture



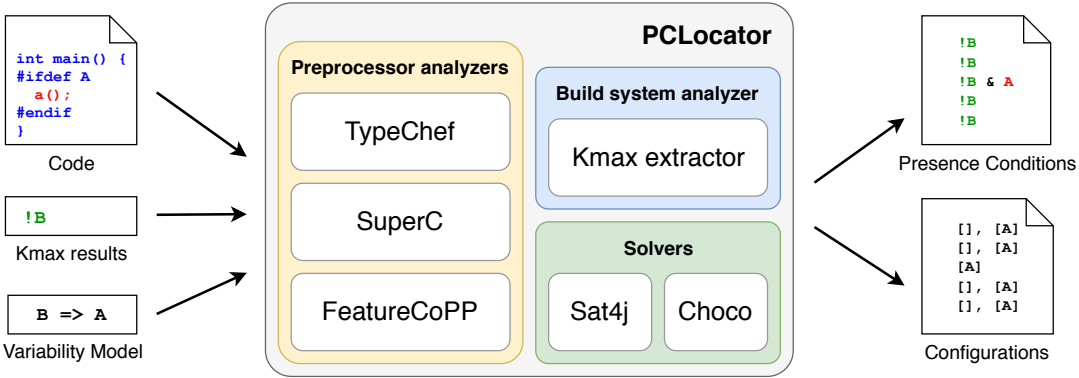
Architecture



Architecture



Architecture



Example

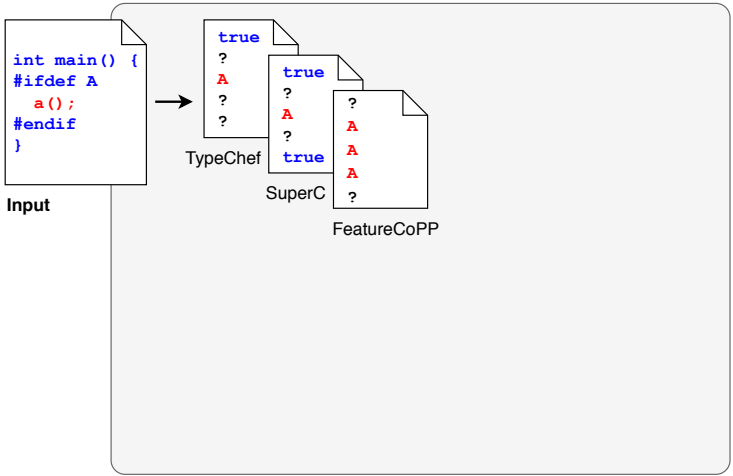
```
int main() {
#ifdef A
  a();
#endif
}
```

Input

Example

Parsing the File

- Parse the file using TypeChef, SuperC and FeatureCoPP
- There are differences between the analyzers



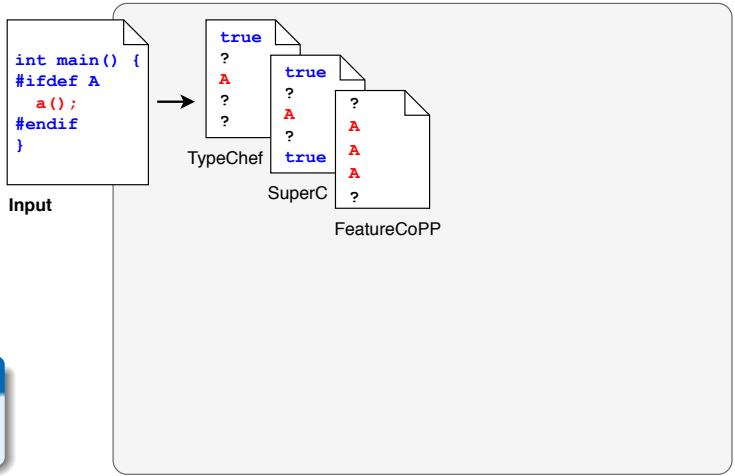
Example

Parsing the File

- Parse the file using TypeChef, SuperC and FeatureCoPP
- There are differences between the analyzers

Conditional Token Stream

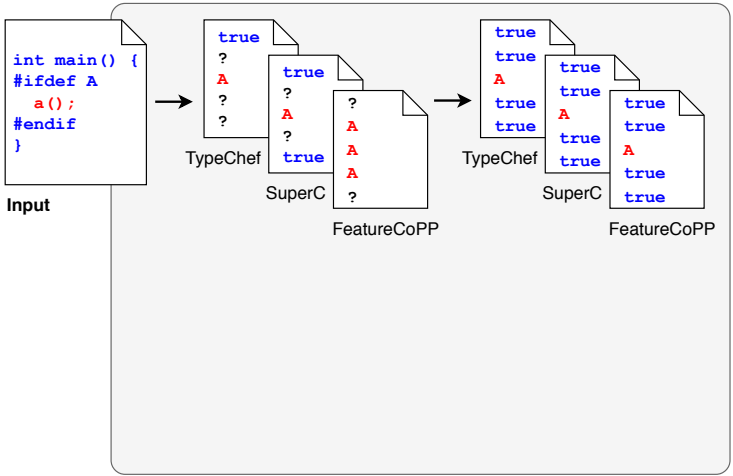
$\texttt{int}_{true}\ \texttt{main}_{true}\ \texttt{a()}_{A}$



Example

Refining the Results

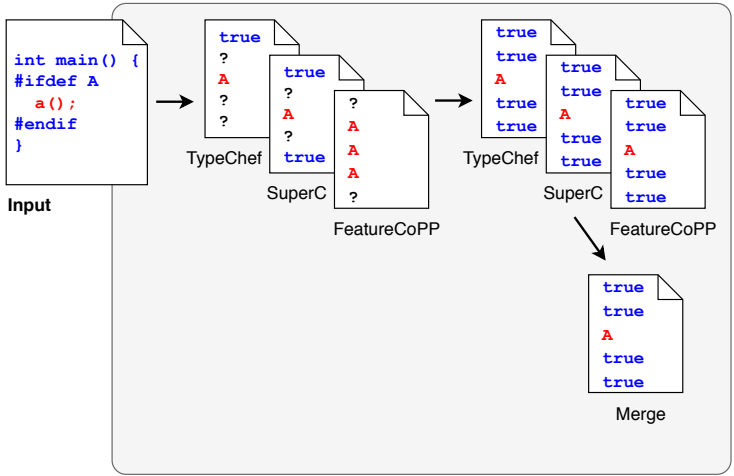
- Deduce missing presence conditions from surrounding lines



Example

Merging the Results

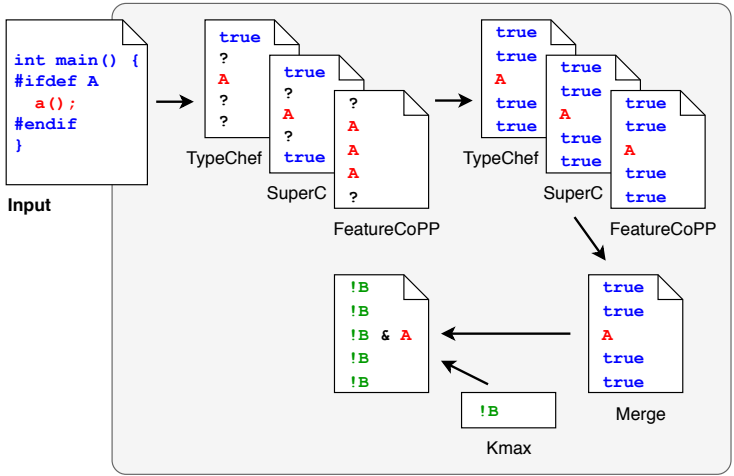
- Choose the "best" presence condition
- TypeChef, SuperC » FeatureCoPP
- Specializations are preferable



Example

Build System analysis

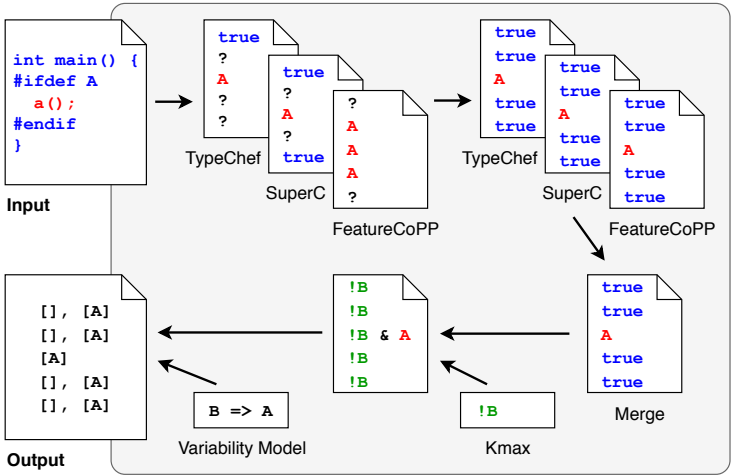
- Consider Kmax results in *Kbuild* projects
- Combine Kmax + Merge presence conditions



Example

Configuration Space

- Call a solver to obtain configurations
- Repeat for more configurations
- Requires a variability model



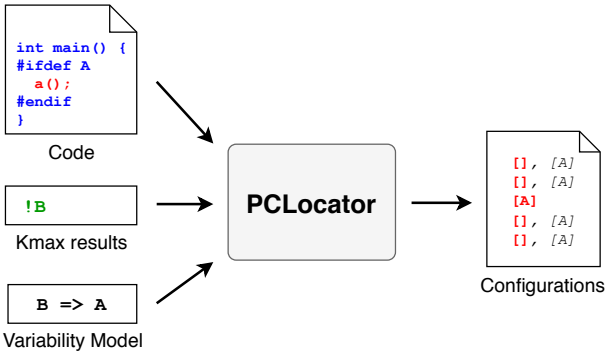
Evaluation Setup

- Target systems:
  - 56 locations in the *Variability Bugs Database* (VBDb)
  - 120 random locations in the *BusyBox toolkit*

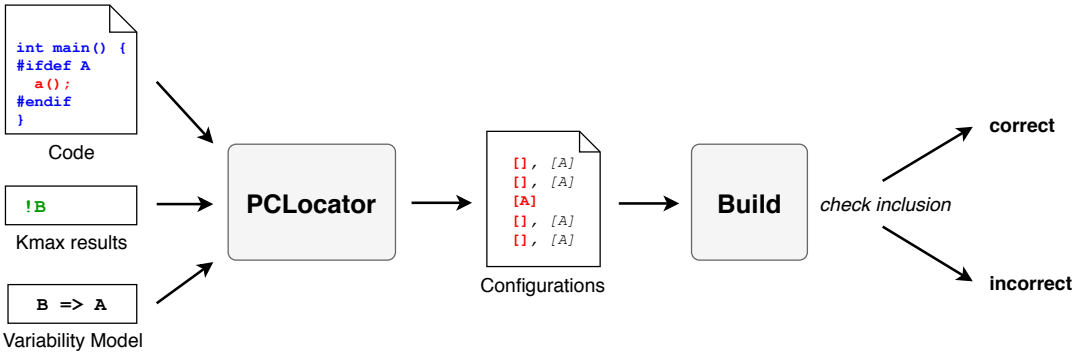
Evaluation Setup

- Target systems:
  - 56 locations in the *Variability Bugs Database* (VBDb)
  - 120 random locations in the *BusyBox toolkit*
- Measured metrics:
  - Analysis time
  - Correctness

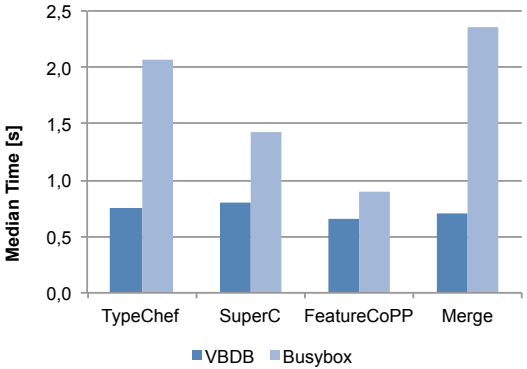
Measuring Correctness



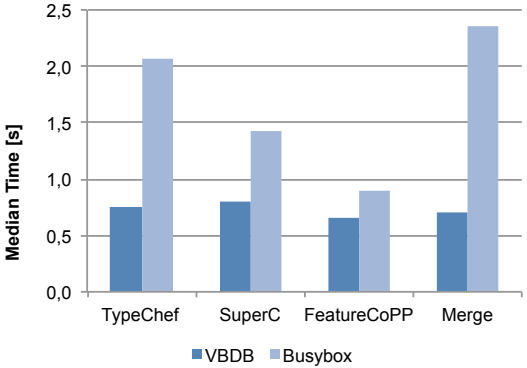
Measuring Correctness

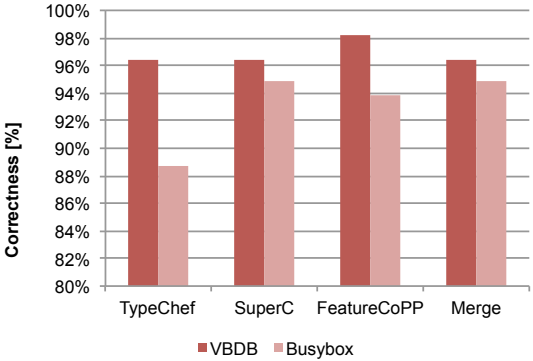


Evaluation

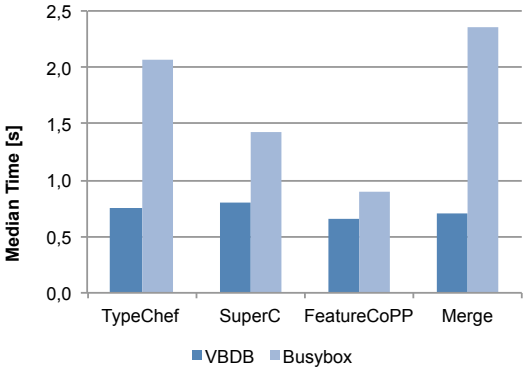

Median Time [s]
2,5
2,0
1,5
1,0
0,5
0,0
TypeChef
SuperC
FeatureCoPP
Merge
VBDB
Busybox

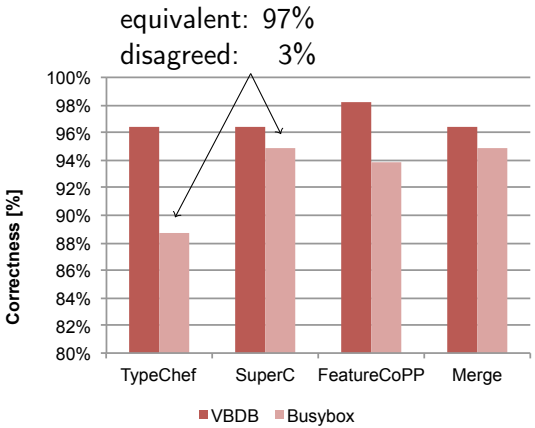
Evaluation


Median Time [s]
2,5
2,0
1,5
1,0
0,5
0,0
TypeChef
SuperC
FeatureCoPP
Merge
VBDB
Busybox


Correctness [%]
100%
98%
96%
94%
92%
90%
88%
86%
84%
82%
80%
TypeChef
SuperC
FeatureCoPP
Merge
VBDB
Busybox

Evaluation

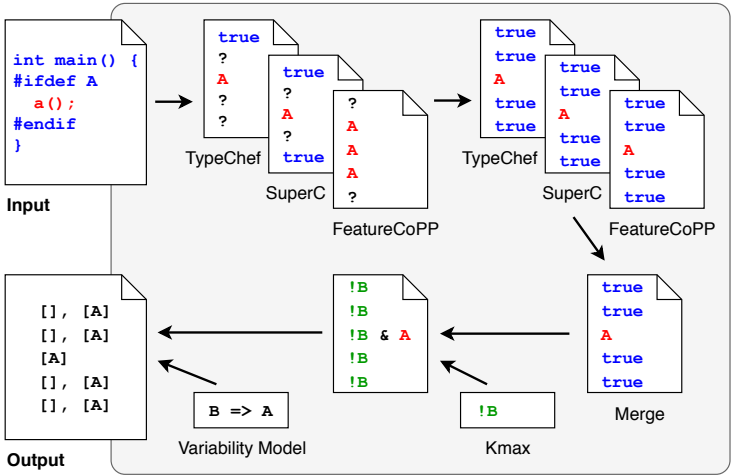

2,5
2,0
1,5
1,0
0,5
0,0
Median Time [s]
TypeChef
SuperC
FeatureCoPP
Merge
VBDB
Busybox


equivalent: 97%
disagreed: 3%
100%
98%
96%
94%
92%
90%
88%
86%
84%
82%
80%
Correctness [%]
TypeChef
SuperC
FeatureCoPP
Merge
VBDB
Busybox

Conclusion

PCLocator Tool Suite

- Obtains Configurations for Code Locations
- Quite accurate results in reasonable time
- freely available: **github.com/ekuiter/PCLocator**



Evaluation Results

VBDb: 56 given locations				
	Merge	**TypeChef**	**SuperC**	**FeatureCoPP**
Time (s)	40.01	42.53	44.62	36.52
Median	0.71	0.76	0.80	0.66
Min	0.62	0.63	0.25	0.10
Max	0.83	0.87	0.90	0.77
Correctness	96%	96%	96%	98%
Precision	96%	98%	96%	98%
Recall	100%	98%	100%	100%

Evaluation Results

BusyBox: 120 random locations				
	Merge	**TypeChef**	**SuperC**	**FeatureCoPP**
Time (s)	277.95	244.60	173.12	110.46
Median	2.35	2.07	1.43	0.90
Min	1.28	1.26	1.00	0.78
Max	3.37	2.81	2.11	1.25
Correctness	95%	89%	95%	94%
Precision	95%	96%	95%	94%
Recall	100%	93%	100%	100%