MANCHESTER 1824
The University of Manchester

LF-checker: Machine Learning Acceleration of BMC for Concurrency Verification

Tong Wu, Edoardo Manino, Fatimah Aljaafari, Pavlos Petoumenos, Lucas Cordeiro

University of Manchester
Department of Computer Science

MANCHESTER 1824
The University of Manchester

Vulnerabilities in multi-threaded programs

```
1  #include <pthread.h>
2  #include <assert.h>
3  int x = 0;
4  void *foo(void* arg) {
5   x++;
6   if (x>1) {
7    x--;
8   }
9   return NULL;
10 }
11
12 int main(void) {
13  pthread_t id1, id2;
14  pthread_create(&id1, 0, foo, 0);
15  pthread_create(&id2, 0, foo, 0);
16  pthread_join(id1, 0);
17  pthread_join(id2, 0);
18  assert(x == 1);
19  return 0;
20 }
```

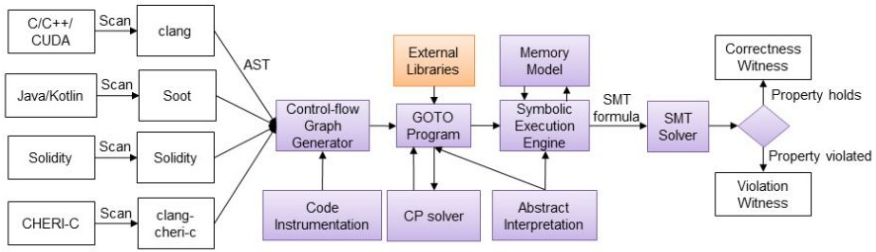
assert(x == 1) could **fail**!

MANCHESTER 1824
The University of Manchester

Bounded Model Checking (BMC) — *ESBMC*

ESBMC: Software Verification Platform

Logic-based automated reasoning for checking the **safety** and **security** of **single- and multi-threaded programs**



Combines BMC, *k*-induction, abstract interpretation, CP/SMT solving towards correctness proof and bug hunting
www.esbmc.org

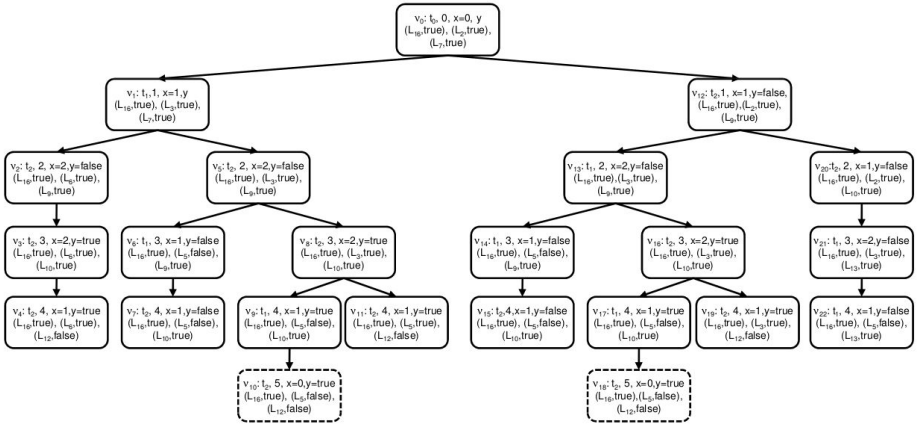
workflow:

- *Control flow graph*
- *goto program convert*
- *symbolic execution*
- *SMT solving*

MANCHESTER 1824
The University of Manchester

Bounded Model Checking (BMC) — *ESBMC for concurrency*

- *Assume Sequential Consistency (SC).*
- *DFS explore all possible executions (interleavings).*



```
1  #include <pthread.h>
2  #include <assert.h>
3  int x = 0;
4  void *foo(void* arg) {
5   x++;
6   if (x>1) {
7    x--;
8   }
9   return NULL;
10 }
11
12 int main(void) {
13  pthread_t id1, id2;
14  pthread_create(&id1, 0, foo, 0);
15  pthread_create(&id2, 0, foo, 0);
16  pthread_join(id1, 0);
17  pthread_join(id2, 0);
18  assert(x == 1);
19  return 0;
20 }
```

MANCHESTER 1824
The University of Manchester

Limit:

The exponentially growing thread interleavings lead the verification process hard to complete.

MANCHESTER 1824
The University of Manchester

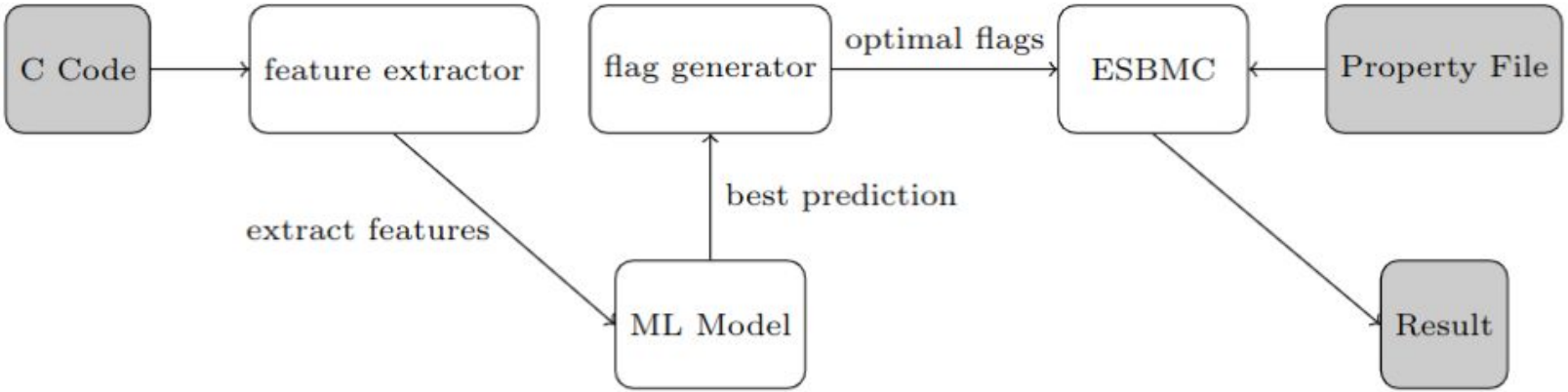
Motivation

- There are different settings in ESBMC for concurrency verification.
- Different combinations of the settings have different overall performance.

- loop unwinding bound
- context switch bound
- partial order reduction
- …

MANCHESTER 1824
The University of Manchester

Workflow of LF-checker



MANCHESTER 1824
The University of Manchester

Stage 1: Verification Engine

- ESBMC exposes a large number of flags that regulate its verification strategy for concurrent program, we list them with their default types and values:

ESBMC Flags	*Type*	*Default*
context-bound	Integer	unlimited
strategy	String	None
k-step	Integer	1
unwind	Integer	unlimited
no-por	Boolean	disabled
no-goto-merge	Boolean	disabled
state-hashing	Boolean	disabled
add-symex-value-sets	Boolean	disabled

MANCHESTER 1824
The University of Manchester

Stage 2: Feature Engineering

- We compute the abstract syntax tree (**AST**) of the program, and extract all possible verification features:

Program Features
threads created
threads joined
mutex locks
atomic locks
global variables access
global function called
binary operators
nondet variables
min. global variable access times
min. global function called times
iteration times in loop

MANCHESTER 1824
The University of Manchester

Stage 3: Training Set Labeling

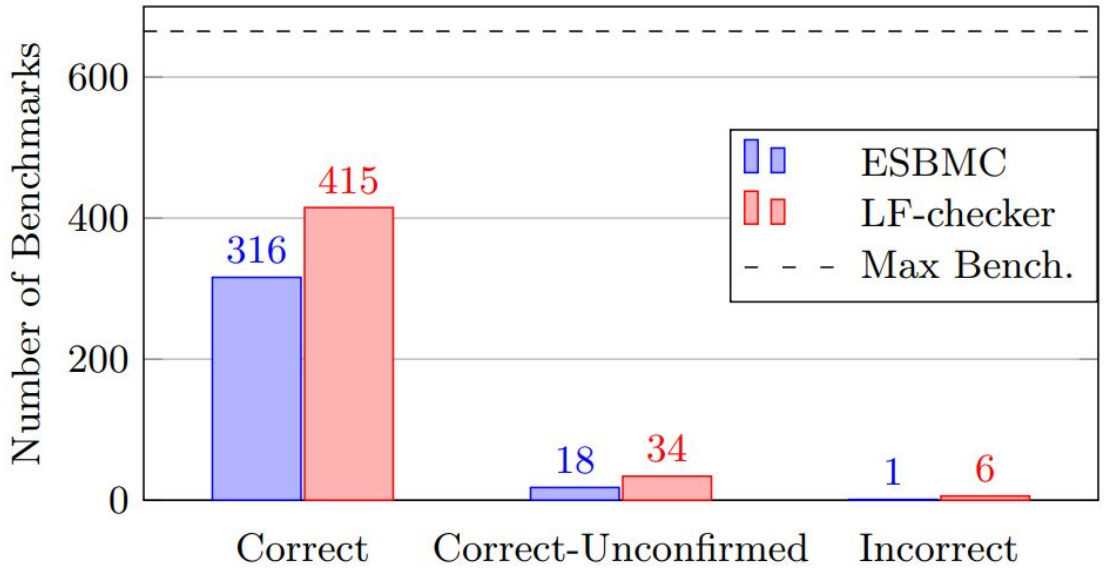
- We use 20% of the benchmarks in SV-COMP (**153** in total) to form train sets.
- Run ESBMC for **3** mins with **240** different combinations of flags.
- Summarise training samples of **36720** with labels ranging from **0** to **5**:

Verification Result	*Class*
correct & time <= 10 (s)	0
correct & 10 (s) < time < 60 (s)	1
correct & time >= 60 (s)	2
unknown	3
timeout	4
incorrect	5

Stage 4: Decision Tree Model

- Decision Tree learns to predict the 0-5 output class given the features of the program and a given choice of flags.
- We try 240 flag combinations and pick the one that yields the lowest output class.

LF-Checker in SV-COMP 2023



We found **context bounds** and **guards merge methods** are key factors affecting the verification

Conculsion: ESBMC can output 34% more correct results compared to using its default settings.

MANCHESTER 1824
The University of Manchester

Limitations

1) The model is only trained on one concurrency category *(unreach-call.ConcurrencySafety*).
2) The model can have better performance after fixing some issues in ESBMC (e.g., guards generation).

Reference

Tong Wu et al. *LF-checker: Machine Learning Acceleration of Bounded Model Checking for Concurrency Verification (Competition Contribution).* 2023. doi: 10.48550 / ARXIV . 2301 . 09142. url: https ://arxiv.org/abs/2301.09142.6

MANCHESTER 1824
The University of Manchester

Thank you