

RACERF: Data Race Detection with Frama-C

SV-COMP'26

Tomáš Dacík^{1,*}

Tomáš Vojnar^{1,2}

¹ Brno University of Technology, Faculty of Information Technology

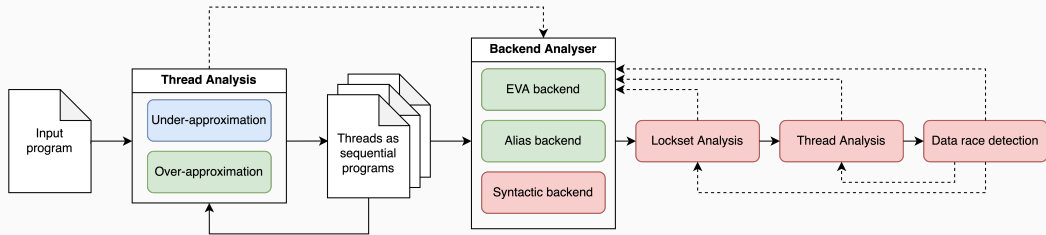
² Masaryk University, Faculty of Informatics

* Supported by Brno Ph.D. Talent scholarship

The RACERF Analyser

- Lightweight static **detection of data races**
- Analysis of concurrent programs using a **backend analyser of sequential programs**
 - Without any modification of backend analyser
 - Interleavings are reflected in **preprocessing** and **initial state** of the analysis
- Heuristic approach **combining under- and over-approximation** + careful ranking
- Implemented as a plugin of the **Frama-C platform**

Analysis Overview

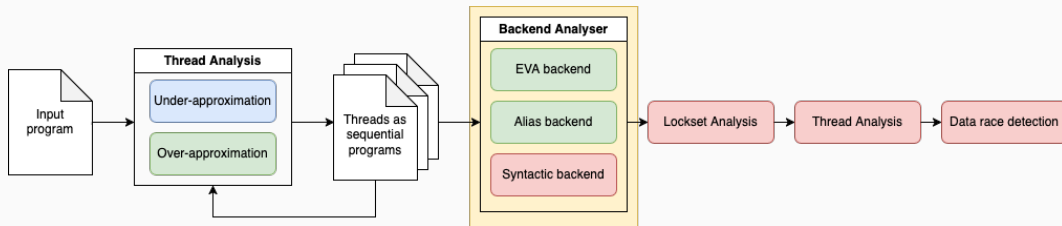


● Under-approximation

● Over-approximation

● No guarantee

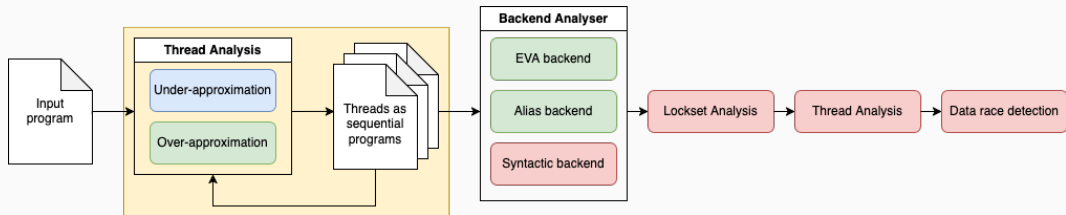
Analysis Overview



Backend analysers of sequential programs

- **EVA** – Frama-C's value analysis (used in SV-COMP)
- **Alias** – Frama-C's alias analysis
- **Syntactic** – fast but imprecise syntactic reasoning

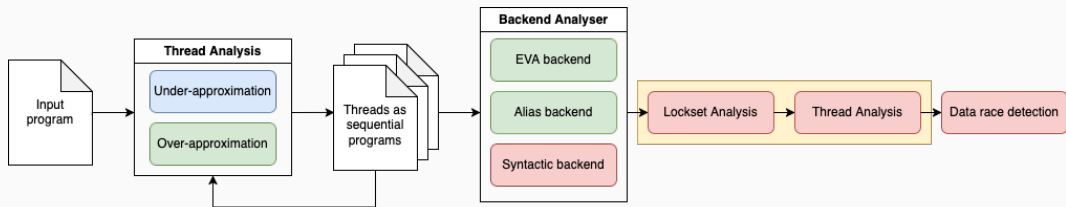
Analysis Overview



Thread Analysis

- Iteratively discovers threads and analyse them as sequential programs
- Each thread is analysed with the initial state given as:
 - **Under-approximation**: join of all states in which it was created
 - **Over-approximation**: join of all discovered states

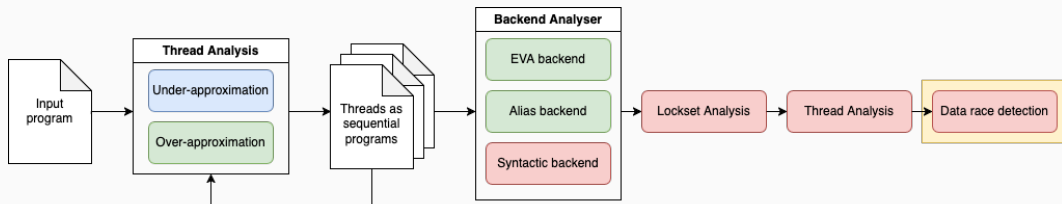
Analysis Overview



May-happen-in-parallel analyses

- Checks whether two analysis contexts can **happen in parallel**
- **Lockset analysis** – lock synchronization
- **Active-thread analysis** – create-join synchronization

Analysis Overview



Data race detection

- Classification to may-/must-races (using results of prev. phases)
- Under-approximation + must-race $\rightsquigarrow$ violation
- Over-approximation + no may-races $\rightsquigarrow$ program claimed correct

Improvements in SV-COMP'26

- Improved **escape analysis**
- Properly implemented **preprocessing for over-approximation**
- Support for **context-sensitive** representation of threads

Improvements in SV-COMP'26

- Improved **escape analysis**
- Properly implemented **preprocessing for over-approximation**
- Support for **context-sensitive** representation of threads
- Added **deadlock detection** to exclude some false alarms:
 - A cyclic dependency in lock-graph
 - A thread does not release a mutex

Results

	Analyser	Correct		Wrong		OOR	Score	Time [s]
		True	False	True	False			
1.	GOBLINT	713	0	0	0	58	1426	94.000
2.	RACERF	679	72	0	2	0	1398	2.900
3.	DEAGLE	676	0	1	0	29	1352	29.000
4.	UGEMCUTTER	607	123	0	0	207	1337	220.000

OOR – out of time/memory, Time – total time (including timeouts)

- **Second place** in *NoDataRace* sub-category
- The only analyser providing (correct) results for *linux-3.14--drivers/** programs (among the most complex in the sub-category)