## Fuzz Testing of Constraint Programming



**Ruben Kindt** 

	REsults
Motivation	
į	
	Found bugs 2 examples
į	
	Place of the bug
į	
į	Type of the bug
'	  found b technique
End motivation	Tround & teermique
Problem I	
ugs are practically unavoidable and always unwanted, especially when a user trusts a program to give a correct	l
and always unwanted, especially when a user trusts a program to give a correct answer and it does not. With solvers surrounding constraint programming languages being executed more and more we would like to strongly avoid any bugs in the real world from arising. To this end it would be interesting to find bugs during development without much overhead, a modern approach would be the use of fuzzers.	
development without much overhead, a modern approach would be the use of fuzzers. which we will try out on a constraint programming language.	
End problem	
	differential Testing
Background	end results
	Terra results
What is CP What is fuzzing	
	Discussion of results
End background	
Approach	
	j
ļ	
  getting seeds	
į	conclusion
Creating CTROM	
!	
Creating Metamorphic tester	
	İ
į	
!	
	Future work
differential Testing	
į	
End approach	Acknowledgements
	Acknowledgements  Although this thesis was finished with a strict planning I put onto myself, it was immensely fun to work on. From the destructive nature of fuzzing bugs to the fascinating topic of constraint solving it all interested me and I had
	Although this thesis was finished with a strict planning I put onto myself, it was immensely fun to work on. From the destructive nature of fuzzing bugs to the fascinating topic of constraint solving it all interested me and I had not a single moment where I had to push myself to start working on it. But this thesis would not have been possible without the following people. Firstly, I would like to thank professor dr. Tias Guns for the guidance and the proposal of this fascinating topic, ir. Ignace Bleukx for answering many questions, intensive thesis meetings, proofreading and the cleverness for coming up with the name of CTORM, dr. ir. Jo Devriendt for finding bugs within our bug finder, the rest of the CPMpy-team, Hakan Kjellerstrand for publishing significant
	meetings, proofreading and the cleverness for coming up with the name of CTORM, dr. ir. Jo Devriendt for finding bugs within our bug finder, the rest of the CPMpy-team, Hakan Kjellerstrand for publishing significant

References
Ruben Kindt, Prof. dr. Tias Guns, and Ir. Ignace Bleukx. "Fuzzing constraint programming languages". Thesis. KU Leuven, Jan. 9, 2023.

Muhammad Numair Mansur et al. "Detecting critical bugs in SMT solvers using blackbox mutational fuzzing". In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2020, pp. 701–712.

further studies.

Meeting on European Software Engineering Conference and Symposium on to Foundations of Software Engineering. 2020, pp. 701–712.

Barton P Miller, Louis Fredriksen, and Bryan So. "An empirical study of the reliability of UNIX utilities". In: Communications of the ACM 33.12 (1990), pp. 32–44.