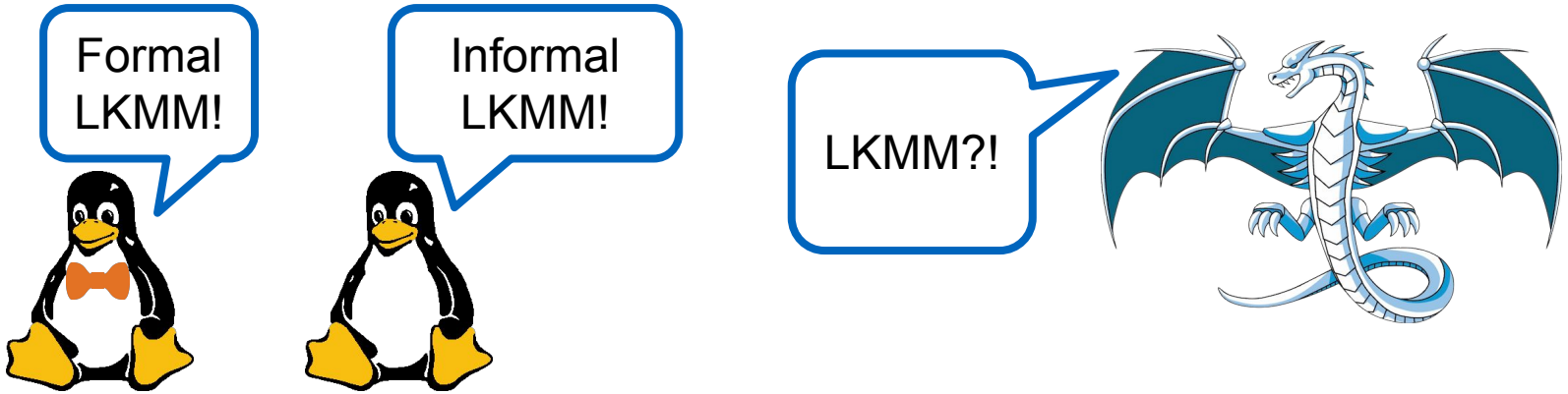
Out-Of-Spec Compilation in the Presence of Dragons: Investigating Broken Dependency Orderings in the Linux Kernel

Paul Heidekrüger
Advisor: Marco Elver



Dependency Orderings are at Risk of Being Broken by Optimizing Compilers

"[W]e are relying on things that are **not guaranteed by the C memory model**, we need to pay attention to the implementations."

- Paul E. McKenney -

"[...] but dammit, **I want to see an actual real example** arguing for why it would be relevant and why the compiler would need our help."

- Linus Torvalds -

Paul McK: https://lore.kernel.org/linux-arm-kernel/20200706173628.GZ9247@paulmck-ThinkPad-P72/
Linus: https://lore.kernel.org/all/CAHk-=wgmUbU6XPHz=4NFoLMxH7j_SR-ky4sKzOBrckmvk5AJow@mail.gmail.com/

What does LKMM even mean?

Frightening Small Children and Disconcerting Grown-ups: Concurrency in the Linux Kernel

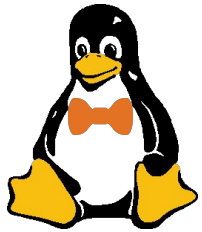
Jade Alglave
University College London
Microsoft Research
j.alglave@ucl.ac.uk

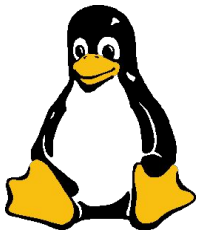
Luc Maranget
Inria — Paris
luc.maranget@inria.fr

Paul E. McKenney
IBM Corporation
Oregon State University
paulmck@linux.vnet.ibm.com

Andrea Parri
Scuola Superiore Sant'Anna
andrea.parri@sssup.it

Alan Stern
Harvard University
stern@rowland.harvard.edu





```
1
2                              ============================
3                              LINUX KERNEL MEMORY BARRIERS
4                              ============================
5
6  By: David Howells <dhowells@redhat.com>
7      Paul E. McKenney <paulmck@linux.ibm.com>
8      Will Deacon <will.deacon@arm.com>
9      Peter Zijlstra <peterz@infradead.org>
```


Alglave, Jade, Luc Maranget, Paul E. McKenney, Andrea Parri, and Alan Stern. "Frightening Small Children and Disconcerting Grown-Ups: Concurrency in the Linux Kernel." In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, 405–18. ACM, 2018. https://doi.org/10.1145/3173162.3177156.

Howells, David, Paul E McKenney, Will Deacon, and Peter Zijlstra. "Memory-Barriers.Txt." Documentation. Linux Kernel Stable Tree, 2006. https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/Documentation/memory-barriers.txt?h=v6.1.53.

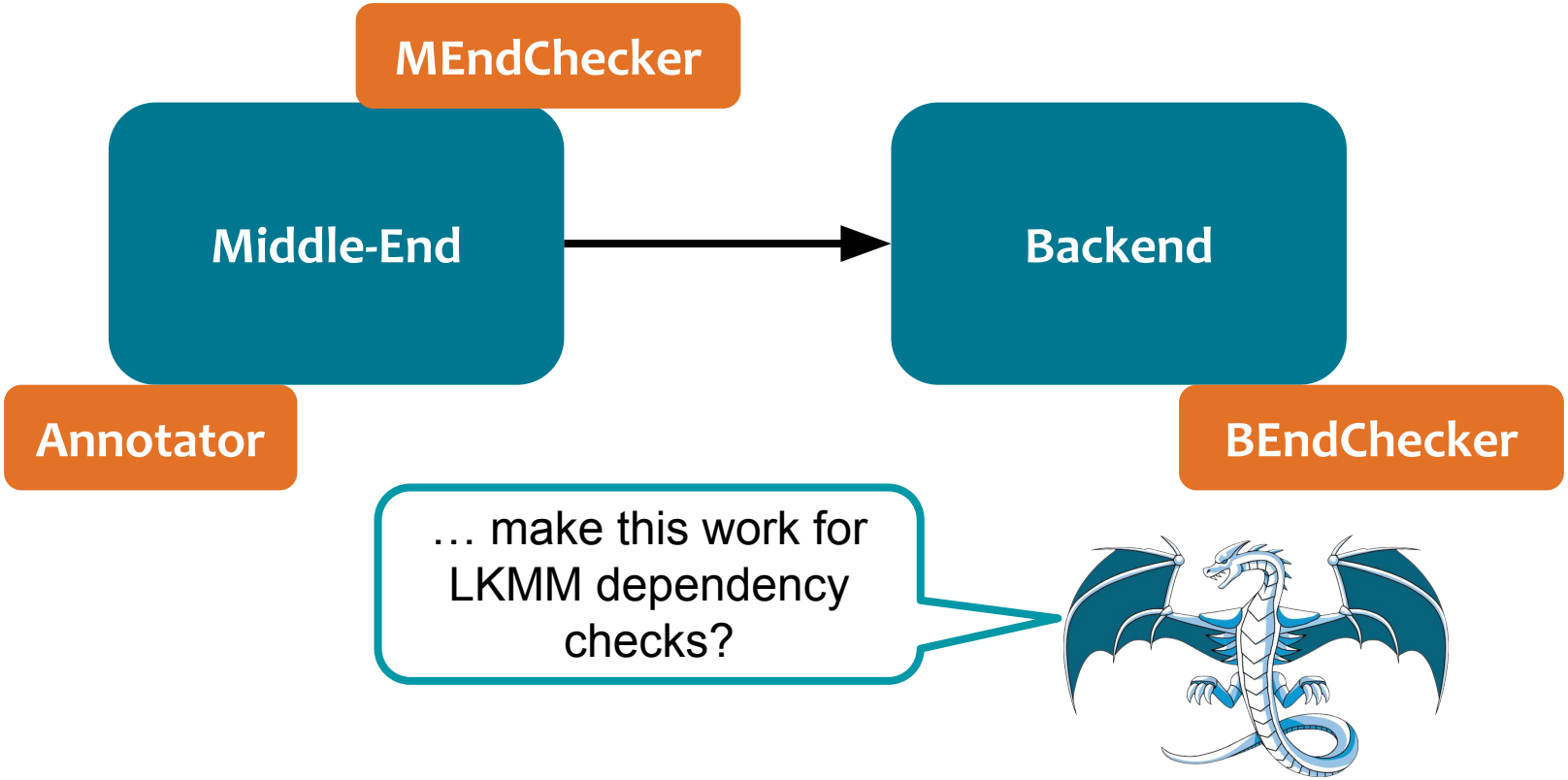
Address Dependencies

```
r1 = READ_ONCE(*foo);
r2 = &r1[42];
r3 = READ_ONCE(*r2);
```

The StatDepChecker

The StatDepChecker

The StatDepChecker



The Limits of Static Analysis

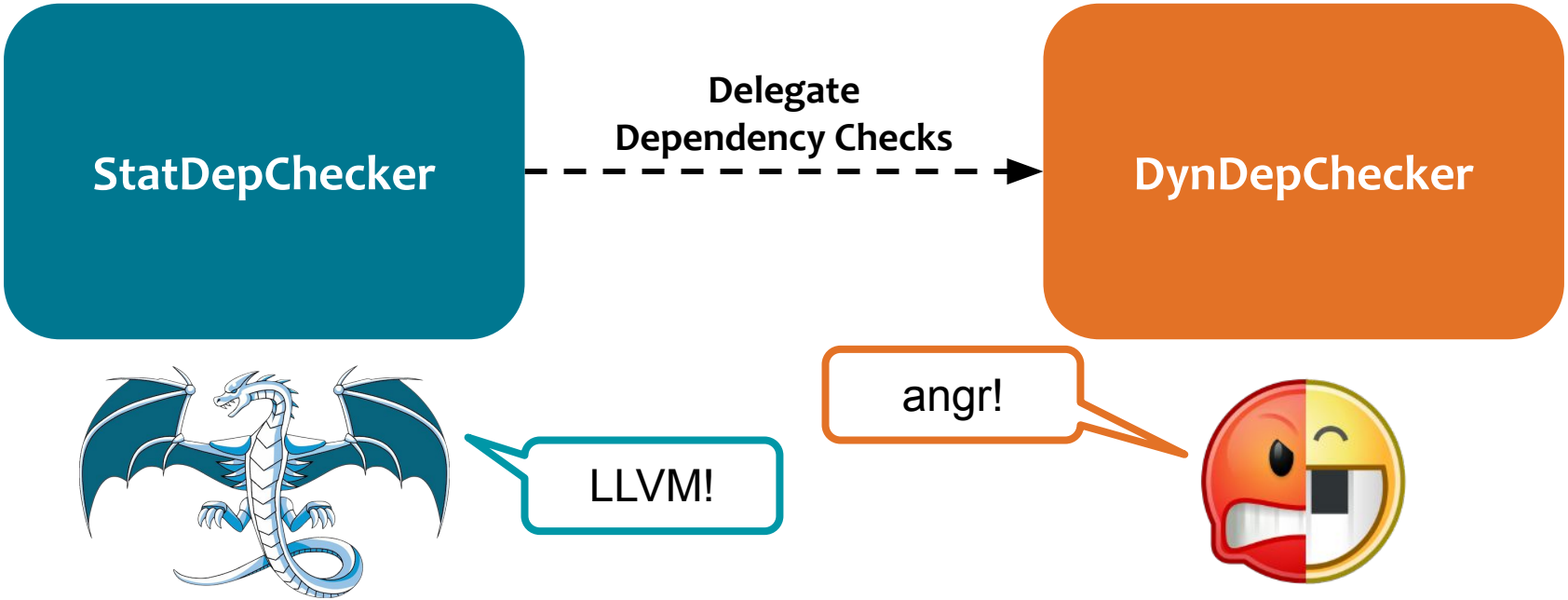
```
mutex_lock(&ksm_thread_mutex); // Lock
```

```
%X = call i32 asm "bswap $0", "=r,r"(i32 %foo) // LLVM Inline Assembly
```

```
%23 = call i1 @llvm.is.constant.i64(i64 %foo) // LLVM Intrinsic
```

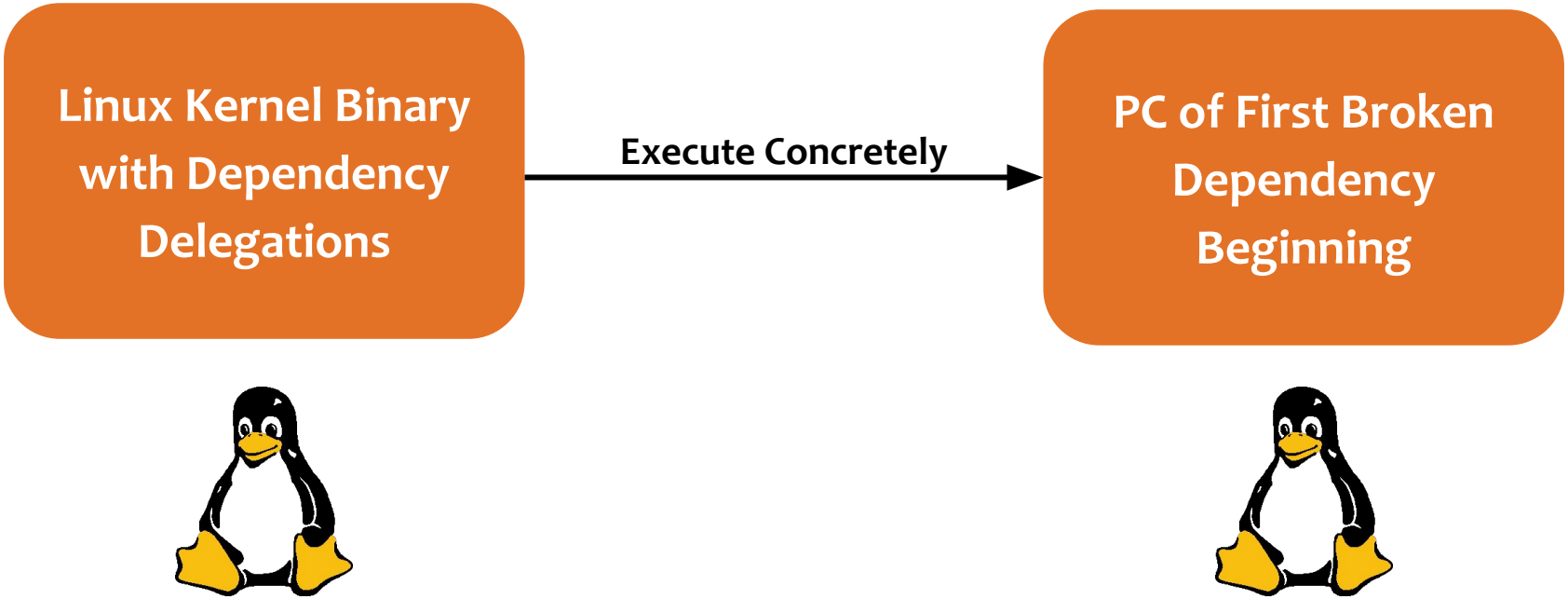
https://llvm.org/docs/LangRef.html

The DynDepChecker

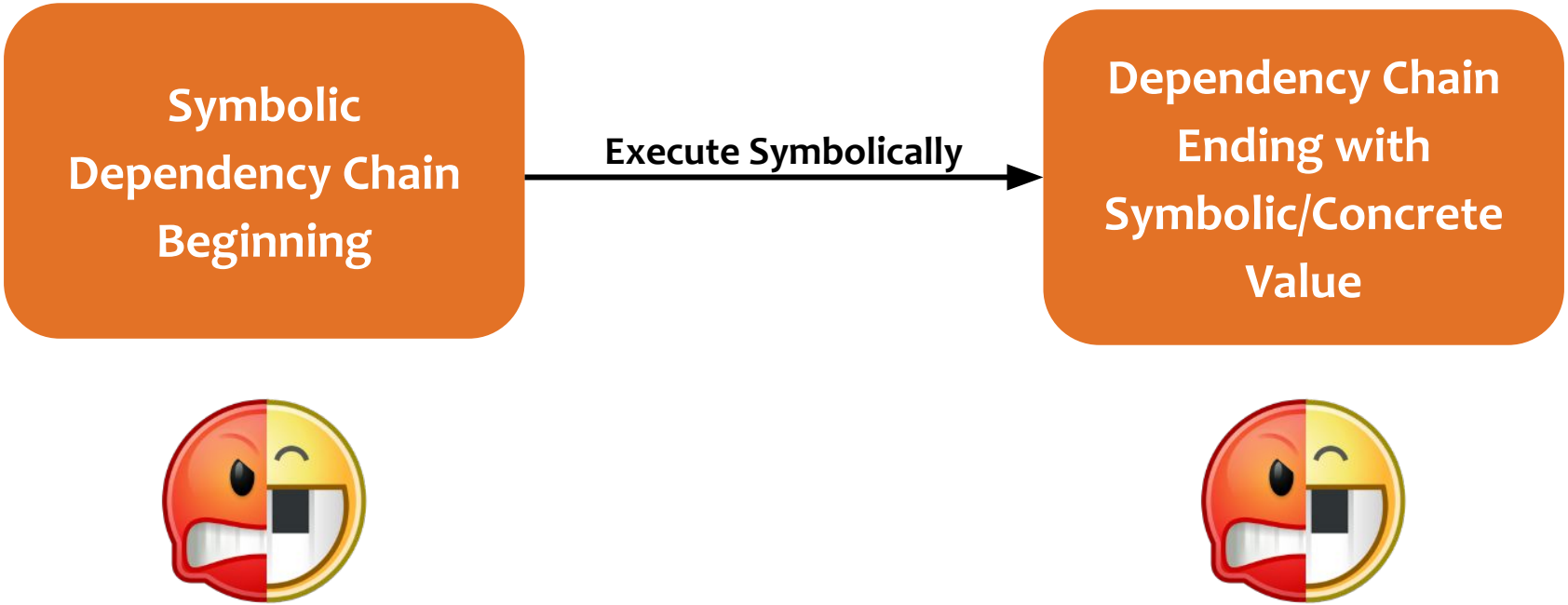


Y. Shoshitaishvili *et al.*, “SOK: (State of) The Art of War: Offensive Techniques in Binary Analysis,” in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 138–157. doi: 10.1109/SP.2016.17.
F. Gritti *et al.*, “SYMBION: Interleaving Symbolic with Concrete Execution,” in *2020 IEEE Conference on Communications and Network Security (CNS)*, Jun. 2020, pp. 1–10. doi: 10.1109/CNS48642.2020.9162164.

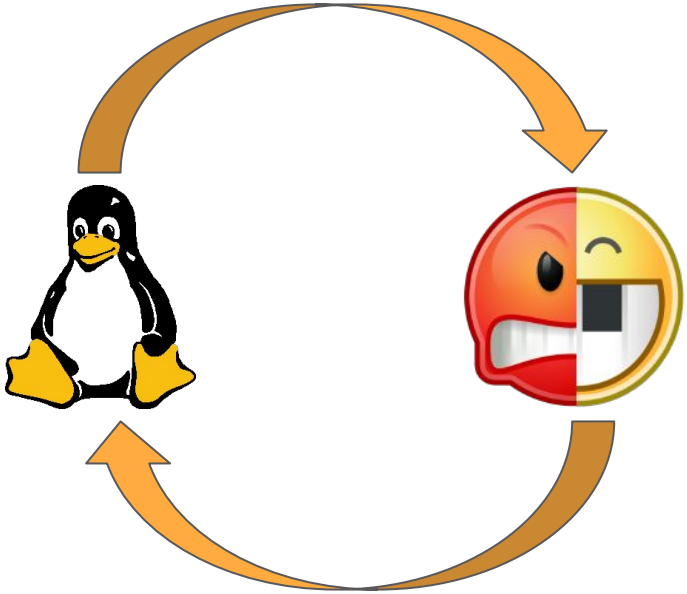
Executing Concretely



Executing Concretely



Interleaved Symbolic Execution



Conclusion and Future Work

- Evaluation
- Complete the StatDepChecker and submit an LLVM RFC
- RCU DepChecker
- Write a paper
- Connect the DynDepChecker to a fuzzer and investigate further use cases
- Strategic lobbying for a dependency annotation mechanism
- Fault-tolerant compiler
- Do a PhD at the