Content:
Triggering systems in modern particle physics experiments often employ short pieces of code in order to make rapid decisions about whether or not incoming data represent potentially interesting physics. Subtle software bugs in triggers are potentially difficult to discover. They can also be very costly as they have the potential to erroneously discard data corresponding to interesting physics. We propose to use formal verification technology to show that triggers behave as expected.

Formal methods provide the highest level of software assurance. More specifically, we propose the development of a domain-specific language for specifying triggers and their properties. This enables the application of formal verification technology to software triggers. The burden of using a domain-specific language is not particularly large, since triggers are generally short programs, and greater software reliability is a substantial payoff considering the importance of the software.

As a first step in this direction, we present a simple example in which trigger-like code is specified and verified using ACL2, an industrial-strength theorem prover. Our long-term goal is to apply such methods to more complex and realistic trigger software.

Primary authors: LEINUNG, A. (); MANOLIOS, P. (); PAUL, T. (); SWAIN, J. (); ATKIN, D. (); COOPERMAN, G. (); KLOPOTOSKI, T. ()
Co-authors: ()
Presenter:
Session classification: --not yet classified--
Track classification: Software Engineering, Data Stores, and Databases
Type: Poster Presentation