inverted CERN School of Computing 2016

Contribution ID: **4** Type: **not specified**

Formal Verification - Robust and Efficient code: Why Formal Verification

Tuesday, 1 March 2016 15:00 (1 hour)

<p>LECTURE 2: In this lecture we will expand on the concepts of the previous lecture and establish formal methods in a broader context, ignoring implementation detail, and investigate how and where these methods are used today, and where they might be used tomorrow. As concrete examples we will be studying how FV can benefit static analysis and comp-cert, and verified C compiler.</p>

<hr>

<p>This talk aims to introduce the concepts of Formal Verification and how they can be used to the benefit of the programmer to produce robust and efficient code. We will be looking into the subject at two levels, both and overview of what FV can concretely bring programmers and going into the nitty-gritty details of theorem proving one of the methods use for FV.</p>

<p>In general, FV means ”proving that certain properties hold for a given system using formal mathematics”. This definition can certainly feel daunting, however, as we will learn, we can reap benefits from the paradigm without digging too deep into the subject.</p>

<p>Examples where FV can help include proving that your code cannot raise division by zero exceptions; produce optimised byte code where the optimisations are proven to be safe and help reason about concurrent systems.</p>

Presenter: ALBERTSSON, Kim (CERN)