If-then-else over the algebra of conditional logic

Gayatri Panicker

The conditional expression `if-then-else` has received considerable importance in programming languages, playing a vital role in the study of program semantics. The juxtaposition of this construct in the context of real-world programming issues leads to the necessity of including a third truth value for the test, viz., `undefined`. For this reason, we also consider program components that themselves might not halt.

Consider the `if-then-else` construct in the most basic sense, i.e.,

$$\texttt{if } \alpha \texttt{ then } s \texttt{ else } t = \begin{cases} s, & \text{if } \alpha = T; \\ t, & \text{if } \alpha = F; \\ \bot, & \text{if } \alpha = U. \end{cases}$$

Here $\bot$ represents the `error` state. A natural question that arises is this: if the tests α came from the three-element C-algebra of McCarthy, then could one give a suitable (possibly complete) axiomatization of this structure?

In this talk we will give an axiomatization for the operation of `if-then-else` over algebras of non-halting programs and non-halting tests by introducing the notion of C-sets where the tests are from a C-algebra. When the C-algebra is an ada, the axiomatization is shown to be complete through a subdirect representation. Further, we will give an axiomatization for the equality test along with `if-then-else` through the notion of agreeable C-sets, which is complete for the class of agreeable C-sets where the C-algebra is an ada. Finally, we will introduce the notion of C-monoids which consider the composition of programs as well as composition of programs with tests along with `if-then-else`. A Cayley-type theorem is obtained in that every C-monoid where the C-algebra is an ada is embeddable in a functional C-monoid.

Keywords
if-then-else, non-halting programs, C-algebra