# Type theory as new constructive foundations for mathematics, logic, and computer science

Jad Issa
American University of Beirut

Thursday, February 11$^{\text{th}}$ 2021, 11 AM

**Abstract**

Since Gödel's incompleteness theorems, interest has generally dropped in foundational mathematics and logic; however, recently, type theory offered a new way to work mathematics and logic that better fits the intuition of mathematicians, is constructive in nature, and fits well with computational reasoning and the design of programming languages and proof assistants. In type theory, proofs are mathematical objects, terms (or elements) of a type representing the corresponding proposition. This leads to mathematical structures (e.g. groups) being defined with the axioms "baked into" the type. Since proofs are treated as objects like any others, the whole logic becomes constructive and can be safely and cleanly automated computationally in proof assistants. This also allows for the flexibility of plugging features in and out of the logic easily. For example, some variants allow for linear logic used in quantum computing and quantitative logic used to control resource use in different contexts including computing, while others such as homotopy type theory, include an $\infty$-groupoid structure on any type and cleanly distinguish propositions, sets, and higher structures as special types and introduce a very powerful 'univalence' axiom that makes isomorphism equivalent to equality reflecting the standard mathematical intuition about the issue. The basic theory, select additions, and relations to category theory and homotopy theory will be discussed alongside some applications.