

# Techniques for Contextual Equivalence in Higher-Order, Typed Languages

Andrew Pitts

University of Cambridge

**Abstract.** Two phrases in a programming language are said to be contextually equivalent if, roughly speaking, they are interchangeable in any complete program without affecting the observable behaviour of the program. I will discuss precise formalisations of this fundamental notion of semantic equivalence for the case of higher-order, typed (HOT) languages, such as ML and Haskell. How does the structure of a type affect properties of contextual equivalence of expressions of that type? It can be very difficult to answer this question when working directly from the definition of contextual equivalence—mainly because HOT programs can make use of their constituent sub-expressions in dynamically complicated ways. This talk will survey some of the semantic techniques (both denotational and operational) that have been devised for proving properties of HOT contextual equivalence.