1042-03-238 **antonio Montalban** (antonio@math.uchicago.edu), Department of Mathematics, 5734 S. University Avenue, Chicago, IL 60637, and **Kenneth A Harris*** (kaharri@umich.edu), Department of Mathematics, 2074 East Hall, 530 Church Street, Ann Arbor, MI 48109-1043.
Back-and-forth invariants for Boolean algebras.

How hard is it to code information into the inherent structure of a Boolean algebra? For example, every low Boolean algebra has a computable copy, so that we cannot code information that is low in computational complexity. On the other hand, a result of Feiner is that there are c.e. Boolean algebras which have no computable copy, although the result does not narrow where on the c.e. hierarchy these sets can lie. An outstanding open question is whether every low$_n$ Boolean algebra has a computable copy; this holds for $n \in \{1, 2, 3, 4\}$, however the argument stalls at low$_5$. What we need to proceed beyond this impasse is a better understanding of the computational complexity inherent in the structure of Boolean algebras. The back-and-forth hierarchy provides a measure of the complexity inherent in the structure of Boolean algebras. We have found invariants for the back-and-forth types occurring at finite levels of the hierarchy. These invariants play a key role in both Feiner's result and the extent proofs that every low$_n$ Boolean algebra has a computable copy (for $n \leq 4$). We will describe these invariants and discuss some of the significant properties of the structure of the back-and-forth types. (Received August 20, 2008)