A Study of the Uniqueness of Source Code

This paper introduced a measurement of uniqueness of source code. It looked at popular C, C++ and Java projects and analyzed their source code. The main focus of the analysis is done by looking at difference granularity defined by tokens which is an atomic part of the source code. The result of the analysis showed that at lower granularity level (approximately 1 line of code) code are not unique but at higher levels things change. Redundancy disappears at around 6 lines for the corpus used by the authors.

Things I would like to see discussed:

- What is the meaning of the result? The authors conducted the study but left out the meaning of the result. The conclusion in the paper seems like a restatement of the result.
- Generizability on context. By looking at the graphs it seems like the results for each project converges to a value and the range of the convergence value have a huge range. I would like to see discussion on this observation.
- Language specific things. I realized this when I was reading the paper, what if the programming language was assembly. It would make no sense to calculate the redundancy at line level for assembly. My question is could the result of uniqueness in some way relates to the whether the programming language is of high or low level? So in an ideal world in the future when code could be automatically generated by describing it in english. It would indeed by unique.