SETL Newsletter # 135B

J.T.Schwartz
January 12, 1975

Additional Thoughts on 'Language Level'

and Optimisation.

1. Manual compilation to lower level involves discovery of, and exploitation of, numerous special facts concerning the context in which particular higher level operations are applied. To the extent that these are irregular and not subject to mechanical discovery, and noting that they are not required for the higher level version, the lower level version of a program P is an *inherently more complex* object than P. Optimisation studies aim to find a subset of the set of optimising transformations which are both stereotyped enough to be applied mechanically and comprehensive enough to compete in optimising effect with the more extensive family of manual devices. To the extent that this succeeds, the lower level language is logically 'absorbed'; to the extent that it fails, the lower level language continues to have an independent existence. Success in absorbing a language level must rest on the ability to discover context-related facts mechanically, and, in particular, to recheck them after program modification. In a manual technique, they would at a minimum have to be listed comprehensively, and transformations not justified by explicitly listed assumptions would have to be suppressed or diagnosed. To do this is probably no easier than to set up a fully mechanical optimiser system.

Even where automatic optimisation fails and a lower level of language continues to remain important for the otherwise unattainable advantages of efficiency it provides, higher level languages will remain important as specification tools.

2. Optimisation problems arising in the reduction to SETL of trans-SETL dictions deserve study. These will include set-theoretic strength reduction and the treatment of converge iterators. Backtracking may be inherently less important, since a backtracking procedure is a 'pruned' exhaustive search and therefore need have no crucial advantage over a simple mathematical description of the object sought.

3. The complexity of processing applied to given inputs is determined by the 'cleverness' of the algorithms used (which will normally have only limited complicating effects), the density of the almost-routine manual optimisation imposed on an underlying abstract algorithm (this may have considerably more serious complicating effects), and finally by the inherent complexity of the input itself. This last factor will generally be large only when linguustic input (describing processes, objects, or logical relationships) is to be handled (another case might be the input and analysis of large tables). Inherent complexity of program should also correlate positively with length of uninterrupted program run. Commericial applications of the ordinary sort will have miscellaneous input, and multiple short runs (against a continuing data base); its complications should therefore almost all arise from manual optimisation, coordination and synchronisation of multiple processes, and real-time considerations.

4. New areas of lingustic processing should arise as formal sematic systems capturing new aspects of the semantic side of natural language processing are understood. Semantic areas worth investigating in this connection are deduction - supplemented fact retrieval, similarity-governed retrieval, and error correction.