APPLICATIVE PROGRAMMING

R.M. Burstall

Dept. of Artificial Intelligence
University of Edinburgh
Edinburgh EH8 9NW

The talk will discuss the advantages and **drawbacks** of the applicative style of programming, also called non-procedural or functional. This stems from "pure LISP", although it is independent of the syntactic peculiarities of LISP, and is well-exemplified by Burge's book "Recursive Programming". The idea is to do without assignment and rely on functions which produce results. Developments for this style include PROLOG and the school of logic programming, (Colmerauer,Kowalski), also the functional programming of Backus which avoids variables altogether, rather in the manner of combinatory logic. The author has recently been experimenting with the design of an applicative language, "HOPE".

The advantages of functional programming include ease of proof and the ability to transform programs easily; in general programs are easy to write and to understand (once you get the trick). Penalties include remoteness from our intuitive idea of things "happening" in the machine and inefficiencies by a factor of n or at least log n in updating data structures.