

DAG Generator

Extend your Compressed Tree generator so that it produces a sequence of DAGs, in which all common sub-expressions have been combined, for a program consisting of a sequence of basic blocks. The target program must be able to include assignment statements, "read", and "write" statements. Each DAG is to continue across basic block boundaries, stopping where control flows converge. Thus, basic block is terminated by a labeled statement, which causes the compiler to forget any information it has about variable values. IF statements are handled carefully, so that (a) the information about variable values known at the end of the <test> is restored at the beginning of the ELSE, and (b) all variable values for variables assigned to in either branch of the IF are killed at the end of the IF. Note that IF statements are hierarchical, so information about assigned-to variables must be propagated from inner IF's to outer. Print the nodes of the completed DAG, showing their node numbers, and interpreting their rule numbers.