Tree-based genetic programming (GP) has several known shortcomings: difficult adaptability to specific programming languages and environments, the problem of closure and multiple types, and the problem of declarative representation of knowledge. Most of the methods that try to solve these problems are based on formal grammars. The precise effect of their distinctive features is often difficult to analyse and a good comparison of performance in specific problems is missing. This thesis reviews three grammar-based methods: context-free grammar genetic programming (CFG-GP), including its variant GPHH recently applied to exam timetabling, grammatical evolution (GE), and LOGENPRO, it discusses how they solve the problems encountered by GP, and compares them in a series of experiments in six applications using success rates and derivation tree characteristics. The thesis demonstrates that neither GE nor LOGENPRO provide a substantial advantage over CFG-GP in any of the experiments, and analyses the differences between the effects of operators used in CFG-GP and GE. It also presents results from a highly efficient implementation of CFG-GP and GE.