USABILITY AND PORTABILITY OF A COMPILER WRITING SYSTEM

Olivier Lecarme
Université de Nice
Nice, France

Abstract

This paper presents a system for automating the production of compilers for programming languages. It emphasizes two particular aspects of this system, its usability and its portability. It also gives a comparison with some good competitive systems, and draws conclusions for future work. The system produces compilers written in the programming language Pascal, from a self-contained description of the language to be compiled. This description uses a notation similar to Backus-Naur form for the syntax, and semantic attributes and actions written in Pascal for the semantics. The system is written entirely in Pascal, and is in current use on several computers.

1 - INTRODUCTION

Since the compiler-compiler of Brooker and Morris (Ros64), numerous systems have been developed for automating part or all of the work that is necessary for producing compilers. Even if the so-called universal programming languages such as PL/I or Algol 68 could be able to have a larger impact on the computer profession than presently, the incredible proliferation of special-purpose programming languages would make necessary such automatic tools, although extensible languages may be considered by some people a viable alternative solution. Feldman and Gries (FeG68) gave a thorough review of the state of the art as of 1967. Since then, no really novel approach has been taken, although many important theories or techniques have been developed, and almost all compiler writing systems* fall

* I prefer this term to "translator writing system", proposed by Feldman and Gries, which seems too much general, and "compiler-compiler" of Brooker and Morris, which designates their own system and approach.

into two categories. In the first one, we find those systems which give an automatic generator for syntactic analyzers (or "parsers"), and almost nothing else (FeH71,McK70,MiS74) ; some users complain that these systems deal only with the easiest part of the job, but it is not completely true : this part is the easiest only because it has been so much well studied that it can be completely automated. In the second category, we find those systems which give a set of languages and processors useful for writing the different parts of a compiler (BSS71,Fel66,Ros64) ; here, the user has to write the compiler himself, but it is much easier than with ordinary programming languages.

Some of these systems have been used for producing effective compilers for useful programming languages, but it is an unfrequent case. The reason for this situation is that the existing compiler writing systems are too often either merely aids to compiler writing, or toys which are not practical for the development of true compilers for actual medium-size languages. As a consequence, they place on the user's shoulders most of the burden of actually programming the compiler he wants. For example, lexical analysis (or "scanning") is often considered an uninteresting job, not worthy of attention from the designers ; error handling and recovery is very often under-estimated, and compilers suffer sad deficiencies in that field ; code generation is judged to be too much machine-dependent to be formalized ; and so on.

In contrast, the system presented in this paper tries to be useful in all aspects of compiler writing, and to be usable for practical languages. The other point I want to emphasize is that this system is claimed to be portable. Portability is now given more and more importance, and it is evident that any large software project such as this one should benefit other people, and not only those who use the computer upon which the system was built. It is especially true when users move from one installation to another one, as I did.

The remaining part of the paper begins with a presentation of the system, sufficient to make the following well understood. The aspects which make the system usable and portable are then discussed in some details. A comparison is made with some important and characteristic competitors, followed by the conclusion.