The Jodie+ programming language

Tibor Csáki
Department of Computer Science
Institute of Informatics
University of Debrecen
H–4010, Debrecen, Pf. 12.
e-mail: csakit@inf.unideb.hu

Abstract

One of the main policies of the Artificial Intelligence (AI) research is the creation of different paradigmatical languages supporting the solutions of AI problems. These developments have resulted the functional LISP and the declarative Prolog programming languages, among others. In spite of its many advantages, the declarative paradigm has the disadvantage of the lack of procedural instruments. These instruments are essential for large and complex development. One possible solution for this problem is the creation and use of hybrid languages.
The usual method of creating a multiparadigmatical language is to extend a declarative one. The results of this method are declarative languages with a mixed structure, but they are not as universal as needed. The other possibility is to create an absolutely new language, regarding that we construct a language which is difficult to learn and not expressive enough.
These problems are why we created this new language Jodie+. The concept of Jodie+ is to establish a link between a declarative (Prolog) and an object oriented (C++) programming language. For this establishment it is necessary to introduce new grammatical elements and make compromises. The main goal of Jodie is to separate the elements of different paradigms. We have solved this requirement with the help of new types which work like functions and we can operate on these special functions using 'conventional' C++ function calls. There are pure Prolog codes in the bodies of these AI functions. The communication has been realised with parameters between the C++ and the Prolog functions.
As a result of these steps, it was obvious to integrate instruments of Automaton Theory (e.g., Turing machine, Lindenmayer System, Finate State Automaton) like we did it with Prolog before. There are also possibilities of integrating other programming objects, as well. To estabilish the usage of automata we introduced new AI constant and AI function types to allow of the description of them.
Since Jodie+ is easy to learn and real Prolog, automaton codes can be placed into it, this language is practical to use in the education of AI and Automaton Theory. Jodie plays on other important role in the field of research, because the C++ frame gives freedom for the scientists.