GUPU: A Prolog course environment and its programming methodology

Ulrich Neumerkel
Institut für Computersprachen
Technische Universität Wien
A-1040 Wien, Austria
ulrich@complang.tuwien.ac.at

Abstract

GUPU is a programming environment specialized for Prolog programming courses which supports a novel way to teaching Prolog. The major improvement in teaching Prolog concerns how programs are read and understood. While the traditional approach covers Prolog's execution mechanism and its relation to mathematical logic we confine ourselves to reading programs informally as English sentences. The student's attention remains focused on a program's meaning instead of details like proof trees or execution traces. Informal reading is limited to short predicates. Larger predicates translate into incomprehensible sentences cluttered with referents and connectives. To overcome this problem a simple reading technique is presented that does not translate the whole predicate at once into English. Only parts of a predicate are considered. The remainder (e.g. some clauses, goals, arguments) is neglected for the moment. In this manner incomprehensible sentences are avoided. Our reading technique extends well to the more procedural aspects of Prolog like termination and resource consumption. The reading technique allows to reason about a program (e.g. understanding, detecting errors) in an efficient static manner avoiding reference to superfluous details of the computation.

GUPU supports this approach by providing a side effect free programming environment. Programs are subject to restrictions which ease informal reading and catch many mostly syntactic and stylistic errors. The cumbersome "type and forget"-style top-level shell is replaced by a side effect free mode of interaction which also improves coding style by allowing to write tests before coding a predicate. The partial evaluator Mixtus is seamlessly integrated into GUPU.

References

[1] A guided tour: http://www.complang.tuwien.ac.at/ulrich/gupu/

[2] U. Neumerkel. Logikorientierte Programmierung, Skriptum, Teil I & II TU-Wien, 1993-1995.