

# Prolog Standardization:

An update

Jonathan Hodgson

Saint Joseph's University

At each of the last four ICLP meetings, WG17, (the ISO working group on Prolog standardization has held a one day meeting) as a workshop. As a result there has been a much greater level of participation at the meetings than had been in many previous years. In spite of this progress has been slow – a common criticism of the ISO process – but there has been progress. Following upon the meeting in Udine it seems appropriate to give the community a sense of where we are with the standard.

To refresh the community's memory the ISO Prolog standard as it exists at present has two parts. The first is the general core and the second modules. The core broke with previous practice in two major areas, it specified a new input output system and a new exception handling system. These features are now found in most if not all Prologs. On the other hand many predicates that are in almost all Prologs (for example `member/2`) are not in the core. The second part modules was much less successful, although WG17 reached a compromise on important issues related to meta-predicates the result is viewed by the community a profoundly unsatisfactory<sup>1</sup>.

With the revival of interest in Prolog standardization dating from the Sitges ICLP the committee has undertaken a number of projects. Specifically there are four projects in varying stages of completion; a technical report on DCGs; a document on predicates not defined in part 1 of the standard but which are generally found in Prolog implementations; a proposal to standardize global variables; and a proposal to standardize threads in Prolog. Current versions of the documents can be found at <http://logtalk.org/plstd/>. (Thanks to Paulo Moura for maintaining this archive.) Let us take a closer look at each of these in turn.

Paulo Moura has been editing the document on DCGs The group now believes that a ballot to approve a document can take place after the next ICLP meeting in July 2009. This optimistic view is a result of the discussions at the Udine meeting. As is often the case in standard writing there is a tension between standardizing on the basis of existing practice and standardization

---

<sup>1</sup>One result of this is that the concept of interface, which could be used as a device for controlling the use of operators in Input Output so that the IO predicates could use different operators for the body of the code, was not taken up at all.

based on ideas as to what ought to be the case. In the case of DCGs one can perhaps characterize the divide based on one's view of the use of `phrase/2,3`. There are those who believe that this is the only proper way to use DCGs and others who do not wish to be bound in this way. Associated to this is the question as to what kinds of things a DCG should process, strictly lists in and lists out or are other things possible

To make this a bit more concrete consider the following examples (due to Peter Szeredi).

```
sum([]) --> [].
sum([X|L]) --> plus(X), sum(L).
```

```
plus(X,S0,S) :- S is S0 + X.
```

```
sum(L,S) :- sum(L,0,S).
```

Purists object to the idea that the non-terminal `plus(X)` is defined by means of a predicate `plus/3`. At some point in the discussion it was noticed that a non terminal `call(G)` would “translate” to `call(G,S0,S1)`. By suitable choice of `G`, assuming the usual (but not yet standardized) `call/3` one could accommodate both the strict list users and other as well. With this observation the group felt confident that the next version of the document will be ready for ballot.

The Japanese Prolog working group has been working on a proposal for backtrackable mutable terms, loosely referred to as “globals”. The document is being edited by Katsuhiko Nakamura who has worked with Nobukuni Kino on an implementation of mutables. The committee feels that there is now general agreement on the issues, with only the semantics of `copy_term/2` to be specified. It is expected that a ballotable proposal will be ready by July.

The standardization of threads is somewhat different from previous work by the committee in that the proposal is being created in parallel with the implementation of threads in several Prologs. In the best case this means that there is less conflicting existing practice.

Finally there is proposed, as what amounts to a large corrigenda to the core, a list of predicates that were “inexplicably omitted” from the original standard. A complete list is to be found in the document `core.pdf` at the above mentioned web site.

With the potential proliferation of parts to the standard the committee is looking into ways in which implementors and users can succinctly specify the level of compliance of to standards of an implementation/system.

It is appropriate to acknowledge here the many people who have contributed to the Prolog standards effort at the recent meetings. I hope this list is comprehensive and apologize to anybody that I may have inadvertently omitted.

With much gratitude to Klaus Daessler , Bart Demoen, Vitor Santos Costa, Pierre Deransart , Joachim Klimpf Paulo Moura , Katsuhiko Nakamura , Ulrich Neumerkel, Roger Scowen ,Peter Szabo, Peter Szeredi ,Markus Triska, Jan Wielemaker, Neng-Fu Zhou.

A special note of thanks is also due to Manuel Hermenegildo, who although he has not been able to attend the WG 17 meetings, has been a source of encouragement for the enterprise. For those who are interested there are pictures of some of these people at <http://www.sju.edu/~jhodgson/wg17/WG17pix.html>