**Port of Marseille Prolog to ODRA 1305 (ICL 1906-compatible)**<http://www.softwarepreservation.org/projects/prolog/#ODRA>  
Paul McJones

"In 1978 we obtained funding for porting Prolog to an ODRA 1305 (essentially an ICL 1900). The machine was much slower, but it had 24-bit words, so there was no question of packing: we had high hopes that the result would be a faster interpreter (in the end it turned out to be twice as fast as on the CYBER). The memory was also only 128K, but we could have all of it, as the machine had only a very simple executive program and was operated in open shop." [[Kluźniak 1984](http://www.softwarepreservation.org/projects/prolog/index.html#Kluzniak1984)]

For a detailed description of this implementation, which is based on the Marseille interpreter, see [[Kluźniak 1984](http://www.softwarepreservation.org/projects/prolog/index.html#Kluzniak1984)].

**Source code**

* Snapshot of ODRA Prolog. Gift of Janusz Bień. [ZIP archive](http://www.softwarepreservation.org/projects/prolog/warsaw/src/odra.zip) [Unpacked](http://www.softwarepreservation.org/projects/prolog/warsaw/src/odra) Kluźniak [[1984](http://www.softwarepreservation.org/projects/prolog/index.html#Kluzniak1984)] notes: "The Prolog system was in the form of four decks of cards."
  + "There was the interpreter proper, which consisted of about 2000 FORTRAN cards." [odra.3](http://www.softwarepreservation.org/projects/prolog/warsaw/src/odra/ODRA.3)
  + "Another FORTRAN program - about 350 cards - was used to create a binary file with the interpreter's internal state. This program, which we called 'The Initiator', could only read in Prolog programs in a very low-level form - essentially a character representation of the internal form of Prolog clauses. The interpreter used Prefix Polish representation of trees, so we called this low-level language Prefix Prolog. To give an example of its distinctive flavour, here is the well-known procedure APPEND(\*):

1.2APPEND3NIL000NIL0  
 3.2APPEND3.2012.203.2APPEND3123NIL0

(\*) The first digit is the number of variables. Each functor is followed by its arity and variables are represented by integer offsets (not ambiguous, as numbers greater than 9 are not allowed)."  
[odra.4](http://www.softwarepreservation.org/projects/prolog/warsaw/src/odra/ODRA.4)

* + "The third deck - about 75 cards - started with a card defining the character set, followed by 17 cards of integer sequences defining the interpreter state's 'kernel' (the representation of NIL, etc.). Seven cards declared the non-character functors and predicates used in the Prefix Prolog program which followed, 'The Bootstrapper', which could read and execute programs written in what we called 'Prolog B'. This was rather primitive, but already similar to full Prolog ('Prolog C'). One could write:

+APPEND(NIL,\*L,\*L)  
 +APPEND(.(\*EL,\*L), \*L2, .(\*EL,\*L3)) -APPEND(\*L,\*L2,\*L3)

" [odra.5](http://www.softwarepreservation.org/projects/prolog/warsaw/src/odra/ODRA.5)

* + "The last deck consisted of about 400 cards in Prolog B, defining the full Prolog Monitor (interpreter with 'real' diagnostics, high-level input/output routines, etc.). The Monitor was written in a style apparently-designed to squeeze the last ounce of advantage from unification's ability to deal with multi-purpose arguments. Despite repeated attempts to read it, we could not at first understand more than small isolated fragments of this program, so for a long time we did it no harm apart from changing French diagnostic messages to Polish." [odra.6](http://www.softwarepreservation.org/projects/prolog/warsaw/src/odra/ODRA.6)