This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

STANDARD

13211-1

First edition
1995-06-01

Information technology — Programming languages — Prolog —

Part 1:
General core

Technologies de l'information — Langages de programmation — Prolog —

Partie 1: Noyau général

Reference number
ISO/IEC 13211-1:1995(E)

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

Contents

Page

Foreword viii

Introduction ix

1 Scope 1
1.1 Notes 1

2 Normative references 1

3 Definitions 2

4 Symbols and abbreviations 10
4.1 Notation 10
4.1.1 Basic mathematical types 10
4.1.2 Mathematical and set operators 10
4.1.3 Other functions 10
4.2 Abstract data type: stack 11
4.3 Abstract data type: mapping 11

5 Compliance 11
5.1 Prolog processor 11
5.2 Prolog text 12
5.3 Prolog goal 12
5.4 Documentation 12
5.5 Extensions 12
5.5.1 Syntax 12
5.5.2 Predefined operators 12
5.5.3 Character-conversion mapping 12
5.5.4 Types 12
5.5.5 Directives 13
5.5.6 Side effects 13
5.5.7 Control constructs 13
5.5.8 Flags 13
5.5.9 Built-in predicates 13

© ISO/IEC 1995
All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.
ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

5.5.10 Evaluable functors . . . 13
5.5.11 Reserved atoms . . . 13

6 Syntax . . . 13
6.1 Notation . . . 13
6.1.1 Backus Naur Form . . . 13
6.1.2 Abstract term syntax . . . 14
6.2 Prolog text and data . . . 15
6.2.1 Prolog text . . . 15
6.2.2 Prolog data . . . 15
6.3 Terms . . . 15
6.3.1 Atomic terms . . . 16
6.3.2 Variables . . . 16
6.3.3 Compound terms – functional notation . . . 16
6.3.4 Compound terms – operator notation . . . 17
6.3.5 Compound terms – list notation . . . 19
6.3.6 Compound terms – curly bracketed term . . . 20
6.3.7 Terms – double quoted list notation . . . 20
6.4 Tokens . . . 20
6.4.1 Layout text . . . 21
6.4.2 Names . . . 21
6.4.3 Variables . . . 23
6.4.4 Integer numbers . . . 23
6.4.5 Floating point numbers . . . 23
6.4.6 Double quoted lists . . . 24
6.4.7 Back quoted strings . . . 24
6.4.8 Other tokens . . . 24
6.5 Processor character set . . . 24
6.5.1 Graphic characters . . . 25
6.5.2 Alphanumeric characters . . . 25
6.5.3 Solo characters . . . 25
6.5.4 Layout characters . . . 25
6.5.5 Meta characters . . . 26
6.6 Collating sequence . . . 26

7 Language concepts and semantics . . . 26
7.1 Types . . . 27
7.1.1 Variable . . . 27
7.1.2 Integer . . . 27
7.1.3 Floating point . . . 28
7.1.4 Atom . . . 29
7.1.5 Compound term . . . 29
7.1.6 Related terms . . . 29
7.2 Term order . . . 30
7.2.1 Variable . . . 31
7.2.2 Floating point . . . 31
7.2.3 Integer . . . 31
7.2.4 Atom . . . 31
7.2.5 Compound . . . 31
7.3 Unification . . . 31
7.3.1 The mathematical definition . . . 31
7.3.2 Herbrand algorithm . . . 31
7.3.3 Subject to occurs-check (*STO*) and not subject to occurs-check (*NSTO*) . . . 33
7.3.4 Normal unification in Prolog . . . 33
7.4 Prolog text . . . 33
7.4.1 Undefined features . . . 34

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

7.4.2 Directives . . . 34
7.4.3 Clauses . . . 35
7.5 Database . . . 36
7.5.1 Preparing a Prolog text for execution . . . 36
7.5.2 Static and dynamic procedures . . . 36
7.5.3 Private and public procedures . . . 36
7.5.4 A logical database update . . . 37
7.6 Converting a term to a clause, and a clause to a term . . . 37
7.6.1 Converting a term to the head of a clause . . . 37
7.6.2 Converting a term to the body of a clause . . . 37
7.6.3 Converting the head of a clause to a term . . . 37
7.6.4 Converting the body of a clause to a term . . . 38
7.7 Executing a Prolog goal . . . 38
7.7.1 Execution . . . 38
7.7.2 Data types for the execution model . . . 38
7.7.3 Initialization . . . 39
7.7.4 A goal succeeds . . . 39
7.7.5 A goal fails . . . 39
7.7.6 Re-executing a goal . . . 39
7.7.7 Selecting a clause for execution . . . 40
7.7.8 Backtracking . . . 40
7.7.9 Side effects . . . 40
7.7.10 Executing a user-defined procedure . . . 40
7.7.11 Executing a user-defined procedure with no more clauses 42
7.7.12 Executing a built-in predicate . . . 42
7.8 Control constructs . . . 43
7.8.1 true/0 . . . 43
7.8.2 fail/0 . . . 43
7.8.3 call/1 . . . 44
7.8.4 !/0 – cut . . . 45
7.8.5 (',')/2 – conjunction . . . 47
7.8.6 (;)/2 – disjunction . . . 47
7.8.7 (->)/2 – if-then . . . 49
7.8.8 (;)/2 – if-then-else . . . 50
7.8.9 catch/3 . . . 51
7.8.10 throw/1 . . . 53
7.9 Evaluating an expression . . . 54
7.9.1 Description . . . 54
7.9.2 Errors . . . 54
7.10 Input/output . . . 54
7.10.1 Sources and sinks . . . 54
7.10.2 Streams . . . 55
7.10.3 Read-options list . . . 58
7.10.4 Write-options list . . . 58
7.10.5 Writing a term . . . 59
7.11 Flags . . . 60
7.11.1 Flags defining integer type *I* . . . 60
7.11.2 Other flags . . . 61
7.12 Errors . . . 61
7.12.1 The effect of an error . . . 62
7.12.2 Error classification . . . 62

8 Built-in predicates . . . 63
8.1 The format of built-in predicate definitions . . . 63
8.1.1 Description . . . 63
8.1.2 Template and modes . . . 64
8.1.3 Errors . . . 64

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

8.1.4 Examples . . . 65
8.1.5 Bootstrapped built-in predicates . . . 65
8.2 Term unification . . . 65
8.2.1 (=)/2 – Prolog unify . . . 65
8.2.2 unify_with_occurs_check/2 – unify . . . 66
8.2.3 (\=)/2 – not Prolog unifiable . . . 67
8.3 Type testing . . . 67
8.3.1 var/1 . . . 67
8.3.2 atom/1 . . . 68
8.3.3 integer/1 . . . 68
8.3.4 float/1 . . . 68
8.3.5 atomic/1 . . . 68
8.3.6 compound/1 . . . 69
8.3.7 nonvar/1 . . . 69
8.3.8 number/1 . . . 69
8.4 Term comparison . . . 70
8.4.1 (@=<)/2 – term less than or equal, (==)/2 – term identical, (\==)/2 – term not identical, (@<)/2 – term less than, (@>)/2 – term greater than, (@>=)/2 – term greater than or equal . . . 70
8.5 Term creation and decomposition . . . 71
8.5.1 functor/3 . . . 71
8.5.2 arg/3 . . . 72
8.5.3 (=..)/2 – univ . . . 72
8.5.4 copy_term/2 . . . 73
8.6 Arithmetic evaluation . . . 74
8.6.1 (is)/2 – evaluate expression . . . 74
8.7 Arithmetic comparison . . . 74
8.7.1 (=:=)/2 – arithmetic equal, (=\=)/2 – arithmetic not equal, (<)/2 – arithmetic less than, (=<)/2 – arithmetic less than or equal, (>)/2 – arithmetic greater than, (>=)/2 – arithmetic greater than or equal . . . 76
8.8 Clause retrieval and information . . . 77
8.8.1 clause/2 . . . 77
8.8.2 current_predicate/1 . . . 78
8.9 Clause creation and destruction . . . 78
8.9.1 asserta/1 . . . 78
8.9.2 assertz/1 . . . 79
8.9.3 retract/1 . . . 80
8.9.4 abolish/1 . . . 81
8.10 All solutions . . . 82
8.10.1 findall/3 . . . 82
8.10.2 bagof/3 . . . 83
8.10.3 setof/3 . . . 84
8.11 Stream selection and control . . . 86
8.11.1 current_input/1 . . . 86
8.11.2 current_output/1 . . . 86
8.11.3 set_input/1 . . . 87
8.11.4 set_output/1 . . . 87
8.11.5 open/4, open/3 . . . 87
8.11.6 close/2, close/1 . . . 88
8.11.7 flush_output/1, flush_output/0 . . . 89
8.11.8 stream_property/2, at_end_of_stream/0, at_end_of_stream/1 . . . 89
8.11.9 set_stream_position/2 . . . 90
8.12 Character input/output . . . 91
8.12.1 get_char/2, get_char/1, get_code/1, get_code/2 . . . 91
8.12.2 peek_char/2, peek_char/1, peek_code/1, peek_code/2 . . . 92

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

8.12.3 put_char/2, put_char/1, put_code/1, put_code/2, nl/0, nl/1 . 94
8.13 Byte input/output . . . 95
8.13.1 get_byte/2, get_byte/1 . . . 95
8.13.2 peek_byte/2, peek_byte/1 . . . 96
8.13.3 put_byte/2, put_byte/1 . . . 97
8.14 Term input/output . . . 98
8.14.1 read_term/3, read_term/2, read/1, read/2 . . . 98
8.14.2 write_term/3, write_term/2, write/1, write/2, writeq/1, writeq/2, write_canonical/1, write_canonical/2 . . . 99
8.14.3 op/3 . . . 101
8.14.4 current_op/3 . . . 102
8.14.5 char_conversion/2 . . . 103
8.14.6 current_char_conversion/2 . . . 103
8.15 Logic and control . . . 104
8.15.1 (\+)/1 – not provable . . . 104
8.15.2 once/1 . . . 105
8.15.3 repeat/0 . . . 105
8.16 Atomic term processing . . . 105
8.16.1 atom_length/2 . . . 106
8.16.2 atom_concat/3 . . . 106
8.16.3 sub_atom/5 . . . 107
8.16.4 atom_chars/2 . . . 108
8.16.5 atom_codes/2 . . . 109
8.16.6 char_code/2 . . . 109
8.16.7 number_chars/2 . . . 110
8.16.8 number_codes/2 . . . 111
8.17 Implementation defined hooks . . . 112
8.17.1 set_prolog_flag/2 . . . 112
8.17.2 current_prolog_flag/2 . . . 112
8.17.3 halt/0 . . . 113
8.17.4 halt/1 . . . 113

9 Evaluable functors . . . 114
9.1 The simple arithmetic functors . . . 114
9.1.1 Evaluable functors and operations . . . 114
9.1.2 Exceptional values . . . 114
9.1.3 Integer operations and axioms . . . 114
9.1.4 Floating point operations and axioms . . . 115
9.1.5 Mixed mode operations and axioms . . . 116
9.1.6 Type conversion operations . . . 117
9.1.7 Examples . . . 117
9.2 The format of other evaluable functor definitions . . . 119
9.2.1 Description . . . 119
9.2.2 Template and modes . . . 119
9.2.3 Errors . . . 119
9.2.4 Examples . . . 119
9.3 Other arithmetic functors . . . 119
9.3.1 (**)/2 – power . . . 119
9.3.2 sin/1 . . . 120
9.3.3 cos/1 . . . 120
9.3.4 atan/1 . . . 120
9.3.5 exp/1 . . . 121
9.3.6 log/1 . . . 121
9.3.7 sqrt/1 . . . 122
9.4 Bitwise functors . . . 122
9.4.1 (>>)/2 – bitwise right shift . . . 122
9.4.2 (<<)/2 – bitwise left shift . . . 122

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

9.4.3 (/\)/2 – bitwise and . . . 123
9.4.4 (\/)/2 – bitwise or . . . 123
9.4.5 (\)/1 – bitwise complement . . . 124

Annex

A Formal semantics . . . 125
A.1 Introduction . . . 125
A.1.1 Specification language: syntax . . . 125
A.1.2 Specification language: semantics . . . 126
A.1.3 Comments in the formal specification . . . 126
A.1.4 About the style of the Formal Specification . . . 127
A.1.5 References . . . 127
A.2 An informal description . . . 127
A.2.1 Search-tree for "pure" Prolog . . . 128
A.2.2 Search tree for "pure" Prolog with cut . . . 131
A.2.3 Search-tree for kernel Prolog . . . 132
A.2.4 Database and database update view . . . 134
A.2.5 Exception handling . . . 135
A.2.6 Environments . . . 135
A.2.7 The semantics of a standard program . . . 136
A.2.8 Getting acquainted with the formal specification . . . 136
A.2.9 Built-in predicates . . . 137
A.2.10 Relationships with the informal semantics of 7.7 and 7.8 138
A.3 Data structures . . . 138
A.3.1 Abstract databases and terms . . . 138
A.3.2 Predicate indicator . . . 143
A.3.3 Forest . . . 143
A.3.4 Abstract lists, atoms, characters and lists . . . 146
A.3.5 Substitutions and unification . . . 148
A.3.6 Arithmetic . . . 149
A.3.7 Difference lists and environments . . . 149
A.3.8 Built-in predicates and packets . . . 150
A.3.9 Input and output . . . 154
A.4 The Formal Semantics . . . 155
A.4.1 The kernel . . . 155
A.5 Control constructs and built-in predicates . . . 170
A.5.1 Control constructs . . . 170
A.5.2 Term unification . . . 171
A.5.3 Type testing . . . 172
A.5.4 Term comparison . . . 172
A.5.5 Term creation and decomposition . . . 173
A.5.6 Arithmetic evaluation - (is)/2 . . . 174
A.5.7 Arithmetic comparison . . . 174
A.5.8 Clause retrieval and information . . . 174
A.5.9 Clause creation and destruction . . . 175
A.5.10 All solutions . . . 178
A.5.11 Stream selection and control . . . 180
A.5.12 Character input/output . . . 183
A.5.13 Byte input/output . . . 189
A.5.14 Term input/output . . . 192
A.5.15 Logic and control . . . 194
A.5.16 Atomic term processing . . . 195
A.5.17 Implementation defined hooks . . . 198

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 13211 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Annex A of this part of ISO/IEC 13211 is for information only.

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

Introduction

This is the first International Standard for Prolog, Part 1 (General Core). It was produced on 20 April 1995.

There is no other International Standard for Prolog.

Prolog (Programming in Logic) combines the concepts of logical and algorithmic programming, and is recognized not just as an important tool in AI (Artificial Intelligence) and expert systems, but as a general purpose high-level programming language with some unique properties.

The language originates from work in the early 1970s by Robert A. Kowalski while at Edinburgh University (and ever since at Imperial College, London) and Alain Colmerauer at the University of Aix-Marseilles in France. Their efforts led in 1972 to the use of formal logic as the basis for a programming language. Kowalski's research provided the theoretical framework, while Colmerauer's gave rise to the programming language Prolog. Colmerauer and his team then built the first interpreter, and David Warren at the AI Department, University of Edinburgh, produced the first compiler.

The crucial features of Prolog are unification and backtracking. Unification shows how two arbitrary structures can be made equal, and Prolog processors employ a search strategy which tries to find a solution to a problem by backtracking to other paths if any one particular search comes to a dead end.

Prolog is good for windowing and multimedia because of the ease of building complex data structures dynamically, and also because the concept of backing out of an operation is built into the language.

Prolog is taught in more UK university computing degrees than any other programming language.

This part of ISO/IEC 13211 defines the general core features of Prolog, and part 2 will define modules.

This is a preview of "ISO/IEC 13211-1:1995". Click here to purchase the full version from the ANSI store.

Information technology — Programming languages — Prolog —

Part 1:
General core

1 Scope

ISO/IEC 13211 is designed to promote the applicability and portability of Prolog text and data among a variety of data processing systems.

This part of ISO/IEC 13211 specifies:

a) The representation of Prolog text,

b) The syntax and constraints of the Prolog language,

c) The semantic rules for interpreting Prolog text,

d) The representation of input data to be processed by Prolog,

e) The representation of output produced by Prolog, and

f) The restrictions and limits imposed on a conforming Prolog processor.

NOTE — This part of ISO/IEC 13211 does not specify:

a) the size or complexity of Prolog text that will exceed the capacity of any specific data processing system or language processor, or the actions to be taken when the corresponding limits are exceeded;

b) the minimal requirements of a data processing system that is capable of supporting an implementation of a Prolog processor;

c) the methods of activating the Prolog processor or the set of commands used to control the environment in which Prolog text is prepared for execution and executed;

d) the mechanisms by which Prolog text is prepared for use by a data processing system;

e) the typographical representation of Prolog text published for human reading;

f) the user environment (top level loop, debugger, library system, editor, compiler etc.) of a Prolog processor.

This part of ISO/IEC 13211 is intended for use by implementors and knowledgeable programmers, and is not a tutorial.

1.1 Notes

Notes in this part of ISO/IEC 13211 have no effect on the language, Prolog text or Prolog processors that are defined as conforming to this part of ISO/IEC 13211. Reasons for including a note include:

a) Cross references to other clauses and subclauses of this part of ISO/IEC 13211 in order to help readers find their way around,

b) Warnings when a built-in predicate as defined in this part of ISO/IEC 13211 has a different meaning in some existing implementations.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 13211. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 13211 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 646 : 1991, *Information technology — ISO 7-bit coded character set for information interchange.*

ISO 2382-15 : 1985, *Data processing — Vocabulary — Part 15: Programming languages.*

ISO 8859-1 : 1987, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.*

ISO/IEC 9899 : 1990, *Programming languages — C.*

ISO/IEC TR 10034 : 1990, *Guidelines for the preparation of conformity clauses in programming language standards.*

ISO/IEC 10967-1 : 1994, *Information technology — Language independent arithmetic — Part 1: Integer and floating point arithmetic.*

BS 6154 : 1981, *Method of defining — Syntactic meta-language.*