INTERNATIONAL STANDARD ISO/IEC 13211-1:1995
TECHNICAL CORRIGENDUM 1

Published 2007-11-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Programming languages — Prolog

Part 1:
General core

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Langages de programmation — Prolog

Partie 1: Noyau général

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 13211-1:1995 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*

NOTE Some text in Mathematical font is expressed using Latex convention, i.e. surrounded with ‘$’ signs.

3.106 mapping

‘mapping’ is used with a second meaning in the standard: add a second definition
A function from a value of one type T to a value of another type R denoted by $T \rightarrow R$

ICS 35.060.00 **Ref. No. ISO/IEC 13211-1:1995/Cor.1:2007(E)**

Published in Switzerland

3.108 most general unifier (MGU)

Replace 'instance' by 'example' because 'instance' is not being used with the meaning defined in 3.95.

3.125 partial list

Replace 'A variable' by 'A *variable*'.
Replace 'second argument' by 'second *argument*'.

3.148 read-term

Replace 'end token.' by 'end token'.

4.1.3.5 Axiom

Replace:
Axiom: if $x > 0$ then $\sqrt{}\ x$ is the positive square root of x else **undefined**.
by
Axiom: if $x \geq 0$ then $\sqrt{}\ x$ is the non-negative square root of x else **undefined**.

6.3.7 Term -- double quoted list notation

If a double quoted list represents an atom (i.e. the Prolog flag 'double_quotes' has value 'atom'), the priority of the term should depend on whether or not the atom is an operator as in 6.3.1.3. ISO/IEC 13211-1 states that the priority of an atom represented by a double quoted list is always zero.
Replace the syntax rule by the four syntax rules:

```
    term =    double quoted list ;
```

Abstract: *l* *dql*
Priority: 0
Condition: Prolog flag `double_quotes` has value `chars`

```
    term =    double quoted list ;
```

Abstract: *l* *dql*
Priority: 0
Condition: Prolog flag `double_quotes` has value `code`

```
    atom =    double quoted list ;
```

Abstract: *a* *dql*
Priority: *n*
Condition: Prolog flag `double_quotes` has value `atom`
Condition: *a* is an operator

```
    atom =    double quoted list ;
```

Abstract: *a* *dql*
Priority: 0
Condition: Prolog flag `double_quotes` has value `atom`
Condition: *a* is not an operator

7.2.5 c) 2)

Replace
2) if XN is the ...
by
2) XN is the ...

7.8.5.4

Replace the first sentence:
Tables 27 and 28 show the execution stack before and after executing the control construct `','(First, second)`.
by
Tables 27 and 28 show the execution stack before and after executing the control construct `','(First, Second)`.

Table 35 line 2

Replace

```
  (else(W), CP)
```

by

```
  (Else, CP)
```

7.8.8.4 last example

Replace

```
  ';'('->'(!,fail), true), true).
```

by

```
  ';'(('->'(!,fail), true), true).
```

7.9.2

Add additional errors:
i) The value of an argument `Culprit` is not a member of the set I

```
     - type_error(integer, Culprit)
```

j) The value of an argument `Culprit` is not a member of the set F

```
     - type_error(float, Culprit)
```

9.1.7 example no. 35 shows these errors are required.

7.12.2 i)

Twice replace
`imp_dep_atom`
by
`Imp_dep_atom`

8.8.1.1 d)

Replace
  Chooses the first element of the list L
by
  Chooses the first element of the list L, unifies it with the term `clause(Head,Body)`

Similarly for f).

8.9.4.1 abolish/1: Description

In the note, replace 'procedures identified' by 'procedure identified'.

8.10.3.4 example no. 20

Replace

```
    [a, b, f(b), f(a)]
```

by

```
    [a, b, f(a), f(b)]
```

8.13.3.4 put_byte/1

Replace

```
put_byte(84).
  If the current output stream contains
    [..., 113,119,101,114]
  Succeeds, and leaves that stream
    [..., 113,119,101,114,116]

put_byte(st_o, 84).
  If the stream associated with st_o contains
    [..., 113,119,101,114]
  Succeeds, and leaves that stream
    [..., 113,119,101,114,116]
```

by

```
put_byte(84).
  If the current output stream contains
    [..., 113,119,101,114]
  Succeeds, and leaves that stream
    [..., 113,119,101,114,84]

put_byte(st_o, 116).
  If the stream associated with st_o contains
    [..., 113,119,101,114]
  Succeeds, and leaves that stream
    [..., 113,119,101,114,116]
```

8.14.1.4 examples no. 2 and 3

Replace

```
st_o
```

by

```
st_i
```

8.14.1.4 example no. 6 (last)

Replace

The current input stream is left with position past-end-of-stream.

by

The current input stream is left in an undefined state.

(Cf. 8.14.1.1 NOTE 2)

8.14.4.1 d)

Replace

Chooses a member of Set_Op and the goal succeeds

by

Chooses a member of Set_Op, unifies it with `(Priority, Op_specifier, Operator)`, and the goal succeeds

8.16.4 atom_chars/2

The sixth example in 8.16.4.4 is

```
atom_chars('North', ['N’ | X]).
  Succeeds, unifying X with
    ['o', ‘r', ‘t', ‘h'].
```

but the procedural description does not permit this.

Replace 8.16.4.1(c) by:

c) Else if `Atom` is an atom whose name is the sequence of characters Seq and `List` unifies with a list `L` such that every element of `L` is the one-char atom whose name is the corresponding element of Seq, then the goal succeeds,

8.16.5 atom_codes/2

The error noted in 8.16.4 implies a similar change in this procedure. Replace 8.16.5.1(c) by:

c) Else if `Atom` is an atom whose name is the sequence of characters Seq and `List` unifies with a list `L` such that every element of `L` is the character code of the corresponding element of Seq, then the goal succeeds,