ISO Prolog, a basis for Prolog extensions

Ulrich Neumerkel
TU Wien, Austria

1,

ISO Prolog, a basis for Prolog extensions

Ulrich Neumerkel
TU Wien, Austria
ISO/IEC JTC1 SC22 WG17 convener

[1,2,]

ISO Prolog, a basis for Prolog extensions

Ulrich Neumerkel
TU Wien, Austria
ISO/IEC JTC1 SC22 WG17 convener

History of Prolog and ISO

— 1972-05

1,2,3,

ISO Prolog, a basis for Prolog extensions

Ulrich Neumerkel
TU Wien, Austria
ISO/IEC JTC1 SC22 WG17 convener

History of Prolog and ISO

— 1972-05 système Œdipe, not yet Prolog, first dif

1,2,3,4,

ISO Prolog, a basis for Prolog extensions

Ulrich Neumerkel
TU Wien, Austria
ISO/IEC JTC1 SC22 WG17 convener

History of Prolog and ISO

— 1972-05 système Œdipe, not yet Prolog, first dif

— 1972 Prolog 0 similar syntax, with dif/2, boum/2
occurs-check option, many cut-like constructs

1,2,3,4,5,

ISO Prolog, a basis for Prolog extensions

Ulrich Neumerkel
TU Wien, Austria
ISO/IEC JTC1 SC22 WG17 convener

History of Prolog and ISO

— 1972-05 système Œdipe, not yet Prolog, first dif

— 1972 Prolog 0 similar syntax, with dif/2, boum/2
occurs-check option, many cut-like constructs

— 1975 Prolog I, no constraints, but errors, cut

1,2,3,4,5,6,

ISO Prolog, a basis for Prolog extensions

Ulrich Neumerkel
TU Wien, Austria
ISO/IEC JTC1 SC22 WG17 convener

History of Prolog and ISO

— 1972-05 système Œdipe, not yet Prolog, first dif

— 1972 Prolog 0 similar syntax, with dif/2, boum/2
occurs-check option, many cut-like constructs

— 1975 Prolog I, no constraints, but errors, cut

— 1977 DEC 10, better syntax, no errors, (mostly) silent failure instead
very influential, basis for ISO, started speed race

— 1981 Fifth Generation Computer Systems, Hungarian MProlog

1,2,3,4,5,6,7,

ISO Prolog, a basis for Prolog extensions

Ulrich Neumerkel
TU Wien, Austria
ISO/IEC JTC1 SC22 WG17 convener

History of Prolog and ISO

— 1972-05 système Œdipe, not yet Prolog, first dif

— 1972 Prolog 0 similar syntax, with dif/2, boum/2
occurs-check option, many cut-like constructs

— 1975 Prolog I, no constraints, but errors, cut

— 1977 DEC 10, better syntax, no errors, (mostly) silent failure instead
very influential, basis for ISO, started speed race

— 1981 Fifth Generation Computer Systems, Hungarian MProlog

— as response, ESPRIT, many Prolog systems, need for standardization

[1,2,3,4,5,6,7,8.]

Standardization

1984 BSI, convener Roger Scowen BS6154 Syntactic metalanguage, EBNF

1,

Standardization

1984 BSI, convener Roger Scowen BS6154 Syntactic metalanguage, EBNF
BSI — AFNOR cooperation

[1,2,]

Standardization

1984 BSI, convener Roger Scowen BS6154 Syntactic metalanguage, EBNF
BSI — AFNOR cooperation
1987 ISO NWI proposal accepted, begin of WG17

1,2,3,

Standardization

1984 BSI, convener Roger Scowen BS6154 Syntactic metalanguage, EBNF
BSI — AFNOR cooperation
1987 ISO NWI proposal accepted, begin of WG17
1995 ISO/IEC 13211-1 published

1,2,3,4,

Standardization

1984 BSI, convener Roger Scowen BS6154 Syntactic metalanguage, EBNF
BSI — AFNOR cooperation
1987 ISO NWI proposal accepted, begin of WG17
1995 ISO/IEC 13211-1 published
Highlights:
+ disambiguated DEC10 syntax (implementations vs. their documentation)
+ unification defined NSTO (not subject to occurs-check), STO undefined
+ multi octet character set handling (MOCSH), characters vs. bytes
+ clean error system, separates instantiation and type/domain errors
– no modules, later 13211-2:2000 weak
– no constraints, but...

1,2,3,4,5,

Standardization

1984 BSI, convener Roger Scowen BS6154 Syntactic metalanguage, EBNF
BSI — AFNOR cooperation
1987 ISO NWI proposal accepted, begin of WG17
1995 ISO/IEC 13211-1 published
Highlights:
+ disambiguated DEC10 syntax (implementations vs. their documentation)
+ unification defined NSTO (not subject to occurs-check), STO undefined
+ multi octet character set handling (MOCSH), characters vs. bytes
+ clean error system, separates instantiation and type/domain errors
– no modules, later 13211-2:2000 weak
– no constraints, but... extension mechanism, 5.5.11

1,2,3,4,5,6.

Extension mechanism

Extensions permitted for many language features, only if

1,

Extension mechanism

Extensions permitted for many language features, only if there is a strictly conforming mode without them (5.1 e)

[1,2,]

Extension mechanism

Extensions permitted for many language features, only if there is a strictly conforming mode without them (5.1 e)
Possible realization with libraries

1,2,3,

Extension mechanism

Extensions permitted for many language features, only if there is a strictly conforming mode without them (5.1 e)
Possible realization with libraries
Including constraints via 5.5.11 Reserved atoms

1,2,3,4,

Extension mechanism

Extensions permitted for many language features, only if
there is a strictly conforming mode without them (5.1 e)
Possible realization with libraries
Including constraints via 5.5.11 Reserved atoms
And thus constraints fit into an ISO conforming system!

1,2,3,4,5,6,

Extension mechanism

Extensions permitted for many language features, only if
there is a strictly conforming mode without them (5.1 e)
Possible realization with libraries
Including constraints via 5.5.11 Reserved atoms
And thus constraints fit into an ISO conforming system!

Previous efforts:

- freeze/2 and frozen/2 — consistency only via labeling
- meta-structures 1988
- attributed variables 1990 — module based

1,2,3,4,5,6,

Extension mechanism

Extensions permitted for many language features, only if
there is a strictly conforming mode without them (5.1 e)
Possible realization with libraries
Including constraints via 5.5.11 Reserved atoms
And thus constraints fit into an ISO conforming system!

Previous efforts:

- freeze/2 and frozen/2 — consistency only via labeling
- meta-structures 1988
- attributed variables 1990 — module based
  - present in SICStus, Scryer; to a lesser degree Ciao, SWI.

1,2,3,4,5,6,7.

Current WG17 work

— DCG in finalization
— Unicode support (based on MOCSH capabilities).
— Prolog prologue — built-ins like length/2.
— dif/2
— clpfd/clpz
— STO-unification — rational trees and beyond
— Queries using answer descriptions (quad)
— Conformity testing

Current WG17 work

— DCG in finalization
— Unicode support (based on MOCSH capabilities).
— Prolog prologue — built-ins like length/2.
— dif/2
— clpfd/clpz
— STO-unification — rational trees and beyond
— Queries using answer descriptions (quad)
— Conformity testing

Your suggestions are welcome!

Current WG17 work

— DCG in finalization
— Unicode support (based on MOCSH capabilities).
— Prolog prologue — built-ins like length/2.
— dif/2
— clpfd/clpz
— STO-unification — rational trees and beyond
— Queries using answer descriptions (quad)
— Conformity testing

Your suggestions are welcome!

Addendum

ISO Prolog works:

`http://www.complang.tuwien.ac.at/ulrich/iso-prolog`
`http://www.complang.tuwien.ac.at/ulrich/iso-prolog#MOCSH`
`http://www.complang.tuwien.ac.at/ulrich/iso-prolog/prologue`
`http://www.complang.tuwien.ac.at/ulrich/iso-prolog/length`
`http://www.complang.tuwien.ac.at/ulrich/iso-prolog/dif`
`http://www.complang.tuwien.ac.at/ulrich/iso-prolog/conformity_testing`

Questions about ISO Prolog:

`https://stackoverflow.com/questions/tagged/iso-prolog`
`https://software.imdea.org/mailman/listinfo/prolog-standard`

[1,2,3].