

termp

Prolog term utility predicates protocol.

author:

Paulo Moura

version:

1.31

date:

2010/6/9

compilation:

static, context_switching_calls

(no dependencies on other files)

Public interface

depth/2

True if the depth of Term is Depth. The depth of atomic terms is zero; the depth of a compound term is one plus the maximum depth of its sub-terms.

compilation:

static

template:

depth(Term, Depth)

mode - number of solutions:

depth(@term, ?integer) - zero_or_one

ground/1

True if the argument is ground.

compilation:

static

template:

ground(Term)

mode - number of solutions:

ground(@term) - zero_or_one

new/1

Creates a new term instance (if meaningful).

compilation:

static

template:

new(Term)

mode - number of solutions:

new(-nonvar) - zero_or_one

occurs/2

True if the variable occurs in the term.

compilation:

static

template:

```
occurs(Variable,Term)
```

mode - number of solutions:

```
occurs(@var,@term) - zero_or_one
```

subsumes/2

The first term subsumes the second term.

compilation:

```
static
```

template:

```
subsumes(General,Specific)
```

mode - number of solutions:

```
subsumes(?term,@term) - zero_or_one
```

subterm/2

The first term is a subterm of the second term.

compilation:

```
static
```

template:

```
subterm(Subterm,Term)
```

mode - number of solutions:

```
subterm(?term,+term) - zero_or_more
```

valid/1

Term is valid.

compilation:

```
static
```

template:

```
valid(Term)
```

mode - number of solutions:

```
valid(@nonvar) - zero_or_one
```

check/1

Checks if a term is valid. Throws an exception if the term is not valid.

compilation:

```
static
```

template:

```
check(Term)
```

mode - number of solutions:

```
check(@nonvar) - one
```

variant/2

Each term is a variant of the other (i.e. they are structurally equivalent).

compilation:

```
static
```

template:

```
variant(Term1,Term2)
```

mode - number of solutions:
variant(@term,@term) - zero_or_one

variables/2

Returns a list of all term variables (ordered as found when doing a depth-first, left-to-right traversal of Term).

compilation:
static

template:
variables(Term,List)

mode - number of solutions:
variables(@term,-list) - one

singletons/2

Returns a list of all term singleton variables (ordered as found when doing a depth-first, left-to-right traversal of Term).

compilation:
static

template:
singletons(Term,Singletons)

mode - number of solutions:
singletons(@term,-list) - one

Protected interface

(none)

Private predicates

(none)