

dictionaryp

Dictionary protocol.

author:

Paulo Moura

version:

2.0

date:

2010/2/26

compilation:

static, context_switching_calls

(no dependencies on other files)

Public interface

as_dictionary/2

Converts a list of key-value pairs to a dictionary.

compilation:

static

template:

as_dictionary(Pairs,Dictionary)

mode - number of solutions:

as_dictionary(@list(pairs),-dictionary) - one

as_list/2

Converts a dictionary to a ordered list of key-value pairs.

compilation:

static

template:

as_list(Dictionary,Pairs)

mode - number of solutions:

as_list(@dictionary,-list(pairs)) - one

clone/3

Clones a dictionary using the same keys but with all values unbound and returning a list of all the pairs in the new clone.

compilation:

static

template:

clone(Dictionary,Clone,ClonePairs)

mode - number of solutions:

clone(+tree,-tree,-list(pairs)) - one

clone/4

Clones a dictionary using the same keys but with all values unbound and returning the list of all pairs in the dictionary and in the clone.

compilation:

static

template:

clone(Dictionary,Pairs,Clone,ClonePairs)

mode - number of solutions:

clone(+tree,-list(pairs),-tree,-list(pairs)) - one

insert/4

Inserts a Key-Value pair into a dictionary, returning the updated dictionary. When the key already exists, the associated value is updated.

compilation:

static

template:

insert(OldDictionary,Key,Value,NewDictionary)

mode - number of solutions:

insert(+dictionary,+ground,@term,-dictionary) - one

delete/4

Deletes a matching Key-Value pair from a dictionary, returning the updated dictionary.

compilation:

static

template:

delete(OldDictionary,Key,Value,NewDictionary)

mode - number of solutions:

delete(+dictionary,@ground,?term,-dictionary) - zero_or_one

update/4

Updates the value associated with Key in a dictionary, returning the updated dictionary. Fails if it cannot find the key.

compilation:

static

template:

update(OldDictionary,Key,NewValue,NewDictionary)

mode - number of solutions:

update(+dictionary,@ground,+term,-dictionary) - zero_or_one

update/5

Updates the value associated with Key in a dictionary, returning the updated dictionary. Fails if it cannot find the key or if the existing value does not match OldValue.

compilation:

static

template:

update(OldDictionary,Key,OldValue,NewValue,NewDictionary)

mode - number of solutions:

update(+dictionary,@ground,?term,+term,-dictionary) - zero_or_one

empty/1

True if the dictionary is empty.

compilation:

static

template:

empty(Dictionary)

mode - number of solutions:

empty(@dictionary) - zero_or_one

lookup/3

Lookups a matching Key-Value pair from a dictionary.

compilation:

static

template:

lookup(Key, Value, Dictionary)

mode - number of solutions:

lookup(+ground, ?term, @dictionary) - zero_or_one

lookup(-ground, ?term, @dictionary) - zero_or_more

previous/4

Returns the previous pair in a dictionary given a key.

compilation:

static

template:

previous(Dictionary, Key, Previous, Value)

mode - number of solutions:

previous(+dictionary, +key, -key, -value) - zero_or_one

next/4

Returns the next pair in a dictionary given a key.

compilation:

static

template:

next(Dictionary, Key, Next, Value)

mode - number of solutions:

next(+dictionary, +key, -key, -value) - zero_or_one

min/3

Returns the pair with the minimum key in a dictionary. Fails if the dictionary is empty.

compilation:

static

template:

min(Dictionary, Key, Value)

mode - number of solutions:

min(+dictionary, -key, -value) - zero_or_one

max/3

Returns the pair with the maximum key in a dictionary. Fails if the dictionary is empty.

compilation:

static

template:

```
max(Dictionary,Key,Value)
```

mode - number of solutions:

```
max(+dictionary,-key,-value) - zero_or_one
```

delete_min/4

Deletes the pair with the minimum key from a dictionary, returning the deleted pair and the updated dictionary.

compilation:

```
static
```

template:

```
delete_min(OldDictionary,Key,Value,NewDictionary)
```

mode - number of solutions:

```
delete_min(+dictionary,-key,-value,-dictionary) - zero_or_one
```

delete_max/4

Deletes the pair with the maximum key from a dictionary, returning the deleted pair and the updated dictionary.

compilation:

```
static
```

template:

```
delete_max(OldDictionary,Key,Value,NewDictionary)
```

mode - number of solutions:

```
delete_max(+dictionary,-key,-value,-dictionary) - zero_or_one
```

keys/2

Returns a list with all dictionary keys.

compilation:

```
static
```

template:

```
keys(Dictionary,List)
```

mode - number of solutions:

```
keys(@dictionary,-list) - one
```

map/2

Maps a closure over each dictionary key-value pair. Fails if the mapped closure attempts to modify the keys.

compilation:

```
static
```

template:

```
map(Closure,Dictionary)
```

meta-predicate template:

```
map(1,*)
```

mode - number of solutions:

```
map(@callable,+dictionary) - zero_or_more
```

map/3

Maps a closure over each dictionary key-value pair, returning the new dictionary. Fails if the mapped closure attempts to modify the keys.

compilation:

static

template:

map(Closure,OldDictionary,NewDictionary)

meta-predicate template:

map(2,*,*)

mode - number of solutions:

map(@callable,+dictionary,-dictionary) - zero_or_more

apply/4

Applies a closure to a specific key-value pair, returning the new dictionary. Fails if the key cannot be found or if the mapped closure attempts to modify the key.

compilation:

static

template:

apply(Closure,OldDictionary,Key,NewDictionary)

meta-predicate template:

apply(2,*,*,*)

mode - number of solutions:

apply(+callable,+dictionary,+key,-dictionary) - zero_or_one

size/2

Number of dictionary entries.

compilation:

static

template:

size(Dictionary,Size)

mode - number of solutions:

size(@dictionary,?integer) - zero_or_one

Protected interface

(none)

Private predicates

(none)