

UTF-8

## Prctica 1

---

Borja Martinena, b190242

---



## Table of Contents

<b>code</b>	<b>1</b>
Usage and interface	1
Documentation on exports	1
bind/1 (prop)	1
binary_byte/1 (pred)	1
hexd/1 (prop)	2
hex_byte/1 (pred)	2
byte/1 (pred)	2
byte_list/1 (pred)	2
hex_to_bin/2 (pred)	3
byte_conversion/2 (pred)	3
byte_list_conversion/2 (pred)	4
tail/3 (pred)	5
get_nth_bit_from_byte/3 (pred)	5
byte_list_clsh/2 (pred)	6
byte_list_crsh/2 (pred)	6
xor/3 (pred)	6
byte_xor/3 (pred)	6
Documentation on imports	6
<b>References</b>	<b>7</b>

## code

The module contains the predicates specified for the first practical exercise of the *Programacin Declarativa* course at ETSIINF in the academic year 2021-2022.

This is mostly composed of predicates dealing with simple byte operations. First binary and hexadecimal digits are defined:

```
bind/1
hexd/1
```

And bytes, which are lists of either 8 bits/binds or 2 hexadecimal digits.

```
binary_byte/1
hex_byte/1
byte/1
```

The rest of the predicates are implemented with these considerations.

## Usage and interface

- **Library usage:**  
:- use\_module(/home/bmcaos/Documents/IngInf/sem6/PROLOG/pr1/code.pl).
- **Exports:**
  - *Predicates:*  
binary\_byte/1, hex\_byte/1, byte/1, byte\_list/1, hex\_to\_bin/2,  
byte\_conversion/2, byte\_list\_conversion/2, tail/3, get\_nth\_bit\_from\_byte/3,  
byte\_list\_clsh/2, byte\_list\_crsh/2, xor/3, byte\_xor/3.
  - *Properties:*  
bind/1, hexd/1.

## Documentation on exports

### bind/1:

PROPERTY

**Usage:** bind(D)  
D is a binary digit.

```
bind(0).
bind(1).
```

### binary\_byte/1:

PREDICATE

**Usage:** binary\_byte(B)  
B is a binary byte. A binary byte is a list of binary digits of length 8.

```
binary_byte([bind(B7), bind(B6), bind(B5), bind(B4),
             bind(B3), bind(B2), bind(B1), bind(B0)]) :-
    bind(B7),
```

```

bind(B6),
bind(B5),
bind(B4),
bind(B3),
bind(B2),
bind(B1),
bind(B0).

```

**hexd/1:**

PROPERTY

Usage: `hexd(D)`

D is a hexadecimal digit.

```

hexd(0).
hexd(1).
hexd(2).
hexd(3).
hexd(4).
hexd(5).
hexd(6).
hexd(7).
hexd(8).
hexd(9).
hexd(a).
hexd(b).
hexd(c).
hexd(d).
hexd(e).
hexd(f).

```

**hex\_byte/1:**

PREDICATE

Usage: `hex_byte(B)`

B is a hexadecimal byte. A binary byte is a list of binary digits of length 8.

```

hex_byte([hexd(H1),hexd(H0)]) :-
    hexd(H1),
    hexd(H0).

```

**byte/1:**

PREDICATE

Usage: `byte(B)`

B is a byte that is: a binary byte or a hexadecimal byte.

```

byte(BB) :-
    binary_byte(BB).
byte(HB) :-
    hex_byte(HB).

```

**byte\_list/1:**

PREDICATE

**Usage:** `byte_list(BL)`

True if BL is a list of valid bytes.

**Other properties:****Test:** `byte_list(BL)`

Un byte hexadecimal.

- *If the following properties hold at call time:*

`BL=[[hexd(a),hexd(b)]]` (= /2)

*then the following properties should hold upon exit:*

`() (BL)` (undefined property)

*then the following properties should hold globally:*

All the calls of the form `byte_list(BL)` do not fail. (not\_fails/1)

**Test:** `byte_list(BL)`

Tres bytes hexadecimal.

- *If the following properties hold at call time:*

`BL=[[hexd(9),hexd(6)],[hexd(5),hexd(c)],[hexd(d),hexd(f)]]` (= /2)

*then the following properties should hold upon exit:*

`() (BL)` (undefined property)

*then the following properties should hold globally:*

All the calls of the form `byte_list(BL)` do not fail. (not\_fails/1)

**Test:** `byte_list(BL)`

Un byte binario.

- *If the following properties hold at call time:*

`BL=[[bind(1),bind(0),bind(0),bind(0),bind(0),bind(1),bind(1),bind(0)]]`  
(= /2)

*then the following properties should hold upon exit:*

`() (BL)` (undefined property)

*then the following properties should hold globally:*

All the calls of the form `byte_list(BL)` do not fail. (not\_fails/1)

**hex\_to\_bin/2:**

PREDICATE

**Usage:** `hex_to_bin(HD,BN)`

True if HD, a hexadecimal digit is equal in value to BN, a binary nibble. Used to convert hex digits to binary.

```
?- hex_to_bin(hexd(b), BN).
BN = [bind(1), bind(0), bind(1), bind(1)] ? ;
no.
```

**byte\_conversion/2:**

PREDICATE

**Usage:** `byte_conversion(HB,BB)`

True if HB, a hex byte, is equal in value to BB, a binary byte. Used to convert hex bytes to binary.

```
?- byte_conversion([hexd(b), hexd(f)], BB).
BB = [bind(1), bind(0), bind(1), bind(1),
      bind(1), bind(1), bind(1), bind(1)] ? ;
no.
```

### Other properties:

**Test:** byte\_conversion(HB,BB)

Byte mximo.

- *If the following properties hold at call time:*

HB=[hexd(f),hexd(f)] (= /2)

*then the following properties should hold upon exit:*

BB=[bind(1),bind(1),bind(1),bind(1),bind(1),bind(1),bind(1),bind(1)] (= /2)

*then the following properties should hold globally:*

All the calls of the form byte\_conversion(HB,BB) do not fail. (not\_fails/1)

**Test:** byte\_conversion(HB,BB)

Byte con valor 1.

- *If the following properties hold at call time:*

HB=[hexd(0),hexd(1)] (= /2)

*then the following properties should hold upon exit:*

BB=[bind(0),bind(0),bind(0),bind(0),bind(0),bind(0),bind(0),bind(1)] (= /2)

*then the following properties should hold globally:*

All the calls of the form byte\_conversion(HB,BB) do not fail. (not\_fails/1)

**Test:** byte\_conversion(HB,BB)

Conversion binario a hexadecimal.

- *If the following properties hold at call time:*

BB=[bind(1),bind(1),bind(1),bind(1),bind(0),bind(0),bind(0),bind(1)] (= /2)

*then the following properties should hold upon exit:*

HB=[hexd(f),hexd(1)] (= /2)

*then the following properties should hold globally:*

All the calls of the form byte\_conversion(HB,BB) do not fail. (not\_fails/1)

### byte\_list\_conversion/2:

PREDICATE

**Usage:** byte\_list\_conversion(HBL,BBL)

True if HBL, a hex byte list, is equal in value to BBL, a binary byte list. Used to convert hex bytes lists to binary.

Two empty lists are considered to be equal in value. Tests with lists longer than length 1 will cause a TeX hbox overflow in trying to display the binary bite list.

```
?- byte_list_conversion([[hexd(b), hexd(f)]], BN).
BN = [[bind(1), bind(0), bind(1), bind(1),
      bind(1), bind(1), bind(1), bind(1)]] ? ;
no.
```

**Other properties:****Test:** `byte_list_conversion(HBL,BBL)`

Regular call with list of length 1.

- *If the following properties hold at call time:*

`HB=[[hexd(4),hexd(4)]]` (= /2)

*then the following properties should hold upon exit:*

`BB=[bind(1),bind(1),bind(1),bind(1),bind(1),bind(1),bind(1),bind(1)]` (= /2)

*then the following properties should hold globally:*

All the calls of the form `byte_list_conversion(HBL,BBL)` do not fail. (not\_fails/1)

**tail/3:**

PREDICATE

**Usage:** `tail(L,R,T)`

Provides a way to access lists in reverse order, instead of accessing them via head/tail, R is the leading part of the list (rest), and T is the last item (tail).

However, this operation has linear complexity, to access lists in reverse order, it would be more efficient to reverse them.

**get\_nth\_bit\_from\_byte/3:**

PREDICATE

**Usage:** `get_nth_bit_from_byte(N,B,BN)`

True if BN is equivalent to the bit at position N of the byte B.

Used to get the bit at position N of the byte where N is a natural number expressed in terms of Peano's axioms, that is, as the successor of another number recursively until reaching 0. Bytes are big endian so the 0th bit is actually the last. This predicate is quite inefficient because it uses the tail/1 predicate, which has linear time complexity, in each recursive step.

**Other properties:****Test:** `get_nth_bit_from_byte(N,B,BN)`

- *If the following properties hold at call time:*

`N=s(s(0))` (= /2)

`B=[hexd(0),hexd(4)]` (= /2)

*then the following properties should hold upon exit:*

`BN=bind(1)` (= /2)

*then the following properties should hold globally:*

All the calls of the form `get_nth_bit_from_byte(N,B,BN)` do not fail. (not\_fails/1)

**Test:** `get_nth_bit_from_byte(N,B,BN)`

- *If the following properties hold at call time:*

`BN=bind(1)` (= /2)

`B=[hexd(0),hexd(8)]` (= /2)

*then the following properties should hold upon exit:*

`N=s(s(s(0)))` (= /2)

*then the following properties should hold globally:*

All the calls of the form `get_nth_bit_from_byte(N,B,BN)` do not fail. (not\_fails/1)



<b>byte_list_clsh/2:</b> <b>Usage:</b> <code>byte_list_clsh(L,CLShL)</code> Performs a circular left shift on byte list L. NOT IMPLEMENTED	PREDICATE
<b>byte_list_crsh/2:</b> <b>Usage:</b> <code>byte_list_crsh(L,CLShL)</code> Performs a circular right shift on byte list L. NOT IMPLEMENTED	PREDICATE
<b>xor/3:</b> <b>Usage:</b> <code>xor(B1,B2,BR)</code> Implements the logical gate xor for two bit B1 and B2 with a result BR. <pre> xor(bind(0),bind(0),bind(0)). xor(bind(0),bind(1),bind(1)). xor(bind(1),bind(0),bind(1)). xor(bind(1),bind(1),bind(0)). </pre>	PREDICATE
<b>byte_xor/3:</b> <b>Usage:</b> <code>byte_xor(B1,B2,B3)</code> Implements a bitwise XOR operation between two input bytes B1 and B2. Hex bytes are first converted to binary bytes and then the xor predicate is used recursively on each bit. B3 is the output byte. Three empty lists <code>B1 = []</code> , <code>B2 = []</code> and <code>B3 = []</code> will validate this predicate. <b>Other properties:</b> <b>Test:</b> <code>byte_xor(B1,B2,B3)</code> <ul style="list-style-type: none"> <li>– <i>If the following properties hold at call time:</i>  <code>B1=[hexd(a),hexd(b)]</code> <span style="float: right;">(= /2)</span>  <code>B2=[hexd(4),hexd(f)]</code> <span style="float: right;">(= /2)</span>  <i>then the following properties should hold upon exit:</i>  <code>B3=[hexd(e),hexd(4)]</code> <span style="float: right;">(= /2)</span>  <i>then the following properties should hold globally:</i>            All the calls of the form <code>byte_xor(B1,B2,B3)</code> do not fail. <span style="float: right;">(not_fails/1)</span> </li> </ul>	PREDICATE

## Documentation on imports

This module has the following direct dependencies:

- *Internal (engine) modules:*  
`term_basic`, `arithmetic`, `atomic_basic`, `basiccontrol`, `exceptions`, `term_compare`, `term_typing`, `debugger_support`, `basic_props`.
- *Packages:*  
`prelude`, `initial`, `condcomp`, `assertions`, `assertions/assertions_basic`, `regtypes`.

## References

(this section is empty)

