## C.5.1 The encoding function

**encvalue**(**in template (value)** any\_typeinpar,

**in** **universal charstring** encoding\_info := "") **return bitstring**

The **encvalue** function encodes a value or template into a bitstring. When the actual parameter that is passed to inpar is a template, it shall resolve to a specific value (the same restrictions apply as for the argument of the **send** statement). The returned bitstring represents the encoded value of inpar, however, the TTCN-3 test system need not make any check on its correctness. The optional encoding\_info parameter is used for passing additional encoding information to the codec.

In addition to the general error causes in clause 16.1.2, error causes are:

* Encoding fails due to a runtime system problem (i.e. no encoding function exists for the actual type of inpar).

## C.5.2 The decoding function

**decvalue**(**inout bitstring** encoded\_value**,**

**out** any\_typedecoded\_value,

**in** **universal charstring** decoding\_info := "") **return integer**

The **decvalue** function decodes a bitstring into a value. The test system shall suppose that the bitstring encoded\_value represents an encoded instance of the actual type of decoded\_value. The optional decoding\_info parameter is used for passing additional decoding information to the codec.

If the decoding was successful, then the used bits are removed from the parameter encoded\_value, the rest is returned (in the parameter encoded\_value), and the decoded value is returned in the parameter decoded\_value. If the decoding was unsuccessful, the actual parameters for encoded\_value and decoded\_value are not changed. The function shall return an integer value to indicate success or failure of the decoding below:

* The return value 0 indicates that decoding was successful.
* The return value 1 indicates an unspecified cause of decoding failure.
* The return value 2 indicates that decoding could not be completed as encoded\_value did not contain enough bits.

The restrictions in clause 16.1.2 apply. If any of these restrictions is applicable, the return value shall be 1.

## C.5.3 The encoding to universal charstring function

**encvalue\_unichar**(**in template** (**value**)any\_typeinpar,

**in** **charstring** string\_encoding := "UTF-8",

**in** **universal charstring** encoding\_info := "")

**return universal charstring**

The **encvalue\_unichar** function encodes a value or template into a universal charstring. When the actual parameter that is passed to inpar is a template, it shall resolve to a specific value (the same restrictions apply as for the argument of the **send** statement). The returned universal charstring represents the encoded value of inpar, however, the TTCN-3 test system need not make any check on its correctness. . The optional encoding\_info parameter is used for passing additional encoding information to the codec. If the optional string\_encoding parameter is omitted, the default value "UTF-8" is used.

The following values (see ISO/IEC 10646 [2]) are allowed as string\_encoding actual parameters:

1. "UTF-8"
2. "UTF-16"
3. "UTF-32"
4. "UCS-2"
5. "UCS-4"

The specific semantics of this function are explained by the following TTCN-3 definition:

**function** encvalue\_unichar(**in** **template**(**value**) any\_type inpar,   
 **in charstring** enc,  
 **in** **universal** **charstring** encoding\_info)

**return** **universal** **charstring** {

**return** oct2unichar(bit2oct(encvalue(inpar, encoding\_info)), enc);

}

In addition to the general error causes in clause 16.1.2, error causes are:

* Encoding fails due to a runtime system problem (i.e. no encoding function exists for the actual type of inpar).
* The given string encoding is not recognized.

## C.5.4 The decoding from universal charstring function

**decvalue\_unichar**(**inout universal charstring** encoded\_value**,**

**out** any\_typedecoded\_value,

**in charstring** string\_encoding := "UTF-8",

**in** **universal** **charstring** decoding\_info := "")

**return integer**

The **decvalue\_unichar** function decodes (part of) a universal charstring into a value. The test system shall suppose that a prefix of the universal charstring encoded\_value represents an encoded instance of the actual type of decoded\_value. The optional decoding\_info parameter is used for passing additional decoding information to the codec.

If the decoding was successful, then the characters used for decoding are removed from the parameter encoded\_value, the rest is returned (in the parameter encoded\_value), and the decoded value is returned in the parameter decoded\_value. If the decoding was unsuccessful, the actual parameters for encoded\_value and decoded\_value are not changed. The function shall return an integer value to indicate success or failure of the decoding below:

* The return value 0 indicates that decoding was successful.
* The return value 1 indicates an unspecified cause of decoding failure.
* The return value 2 indicates that decoding could not be completed as encoded\_value did not contain enough bits.

If the optional string\_encoding parameter is omitted, the default value "UTF-8" is used.

The following values (see ISO/IEC 10646 [2]) are allowed as string\_encoding actual parameters:

1. "UTF-8"
2. "UTF-16"
3. "UTF-32"
4. "UCS-2"
5. "UCS-4"

The semantics of the function can be explained by the following TTCN-3 function:

**function** decvalue\_unichar  
 (**inout** **universal** **charstring** encoded\_value,   
 **out** any\_type decoded\_value,

**in** **charstring** string\_encoding := "UTF-8",

**in** **universal** **charstring** decoding\_info := "") **return** **integer** {

**var** **bitstring** str = oct2bit(unichar2oct(encoded\_value, string\_encoding));

**var** **integer** result := decvalue(str, decoded\_value, decoding\_info);

**if** (result == 0) { // success

encoded\_value := oct2unichar(bit2oct(str), string\_encoding);

}

**return** result;

}

The restrictions in clause 16.1.2 apply. If any of these restrictions is applicable or if the given string\_encoding is not recognized, the return value shall be 1.