

ASN.1: encoding

Gusztáv ADAMIS

BME TMIT

adamis@tmit.bme.hu

György RÉTHY, János Zoltán SZABÓ

Test Competence Center, Ericsson Hungary

Tuesday, February 21, 2017

Contents

BER

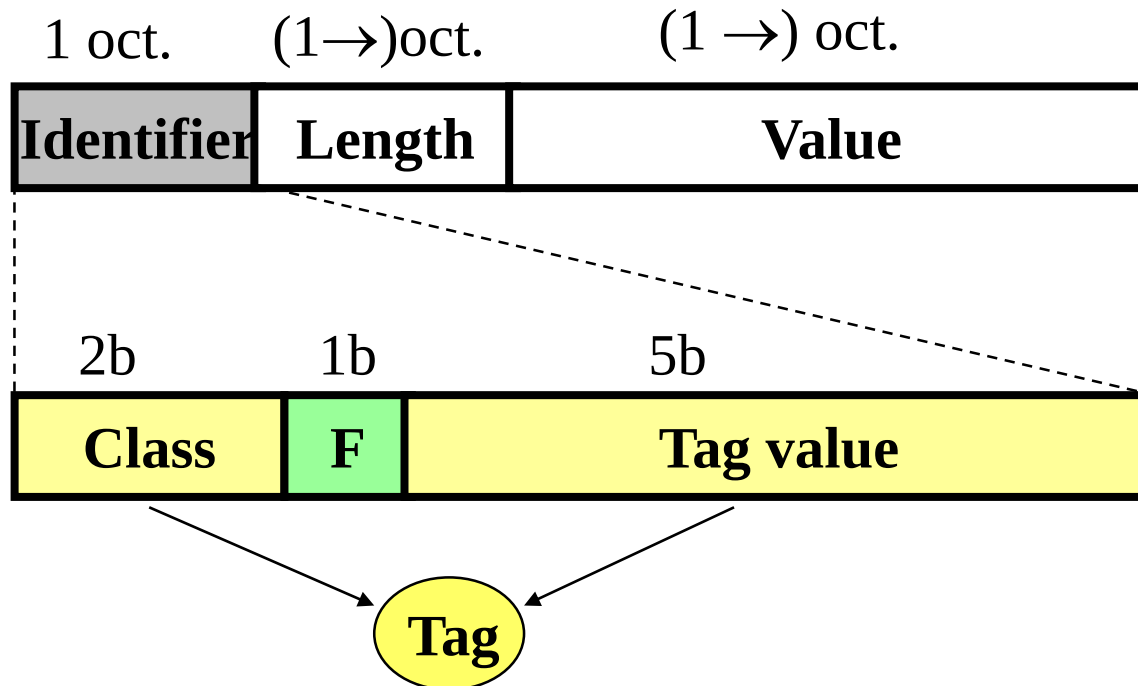
Tagging

Tag of ASN.1 types

Encoding of different types

BER:**Basics of Coding**

- **BER IS ALWAYS OCTET-ORIENTED AND OCTET ALIGNED!**
- **Uses a TLV (type-length-value) structure**



Class:
00 - UNIVERSAL
01 - APPLICATION
10 - context specific
11 - PRIVATE

Format (P/C)
0 - primitive
1 - constructed

Most Important Universal TAG Values

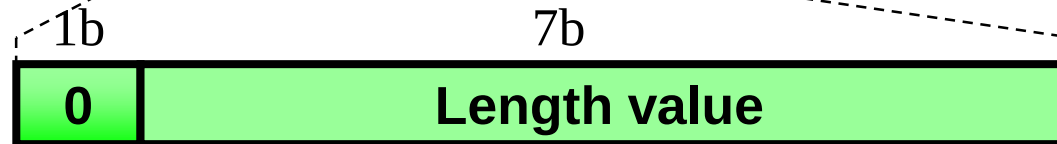
BOOLEAN	1
INTEGER	2
BIT STRING	3
OCTET STRING	4
NULL	5
ENUMERATED	10
SEQUENCE	16
SEQUENCE OF	16
SET	17
SET OF	17

ASN.1 abstract syntax

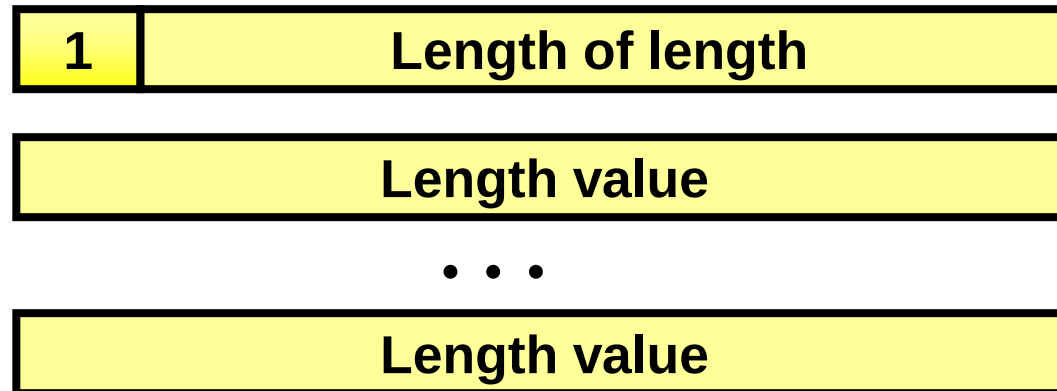
UNIVERSAL 0	<i>Reserved for use by the encoding rules</i>
UNIVERSAL 1	BOOLEAN
UNIVERSAL 2	INTEGER
UNIVERSAL 3	BIT STRING
UNIVERSAL 4	OCTET STRING
UNIVERSAL 5	NULL
UNIVERSAL 6	OBJECT IDENTIFIER
UNIVERSAL 7	ObjectDescriptor
UNIVERSAL 8	EXTERNAL INSTANCE OF
UNIVERSAL 9	REAL
UNIVERSAL 10	ENUMERATED

UNIVERSAL class tags

UNIVERSAL 11	EMBEDDED PDV
UNIVERSAL 12	UTF8String
UNIVERSAL 13	Relative OID
UNIVERSAL 14-15	<i>Reserved</i>
UNIVERSAL 16	SEQUENCE SEQUENCE OF
UNIVERSAL 17	SET SET OF
UNIVERSAL 18-22	Character string types
UNIVERSAL 23-24	Time types
UNIVERSAL 25-30	Character string types
UNIVERSAL 31-...	<i>Reserved</i>

BER:**Coding of the length field****Length ALWAYS counts in octets!****Short form**

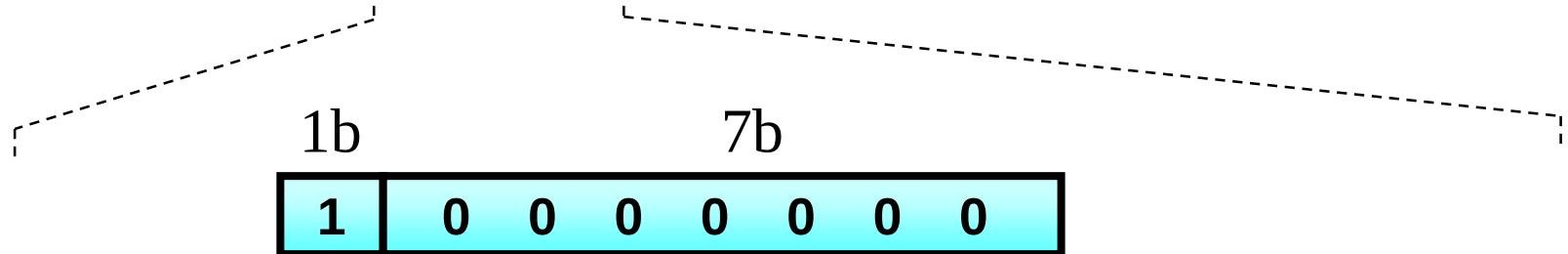
0-127

Long form

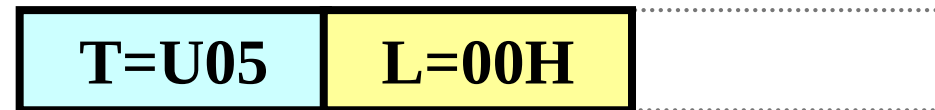
0-2¹²⁶
 ↑ ↑
 (LL=127 is reserved!)

BER:**Coding of the length field**

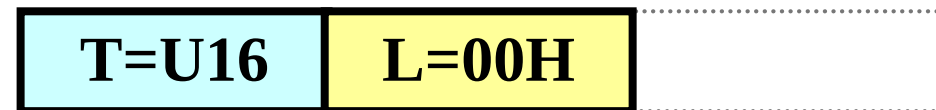
- For **constructed** types only!

**Indefinite form**

Decoded as UNIVERSAL zero type (reserved),
primitive encoded, short form length of value = 0

BER:***When the length is null*****sample-1 NULL ::= NULL**

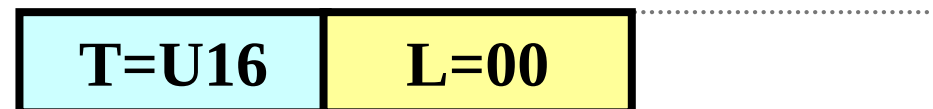
**sample-2 SEQUENCE{
 first INTEGER OPTIONAL,
 second BOOLEAN OPTIONAL} ::= {}**

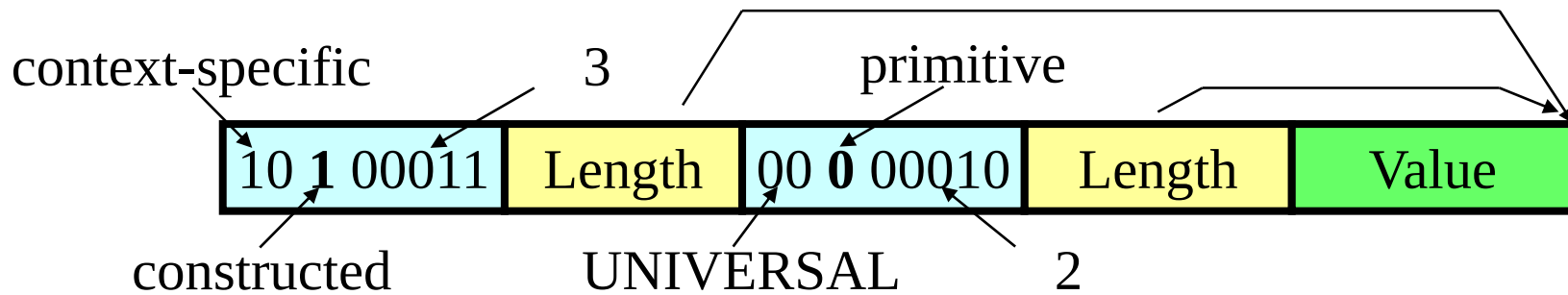
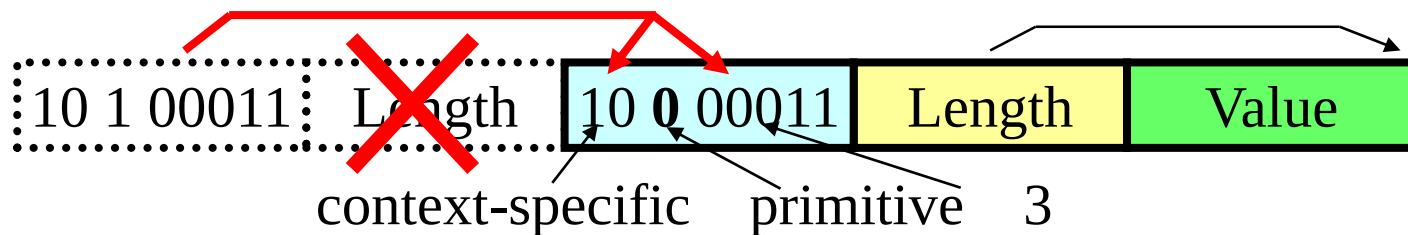


BUT!

T=U16	LL=81H	L=00H
-------	--------	-------

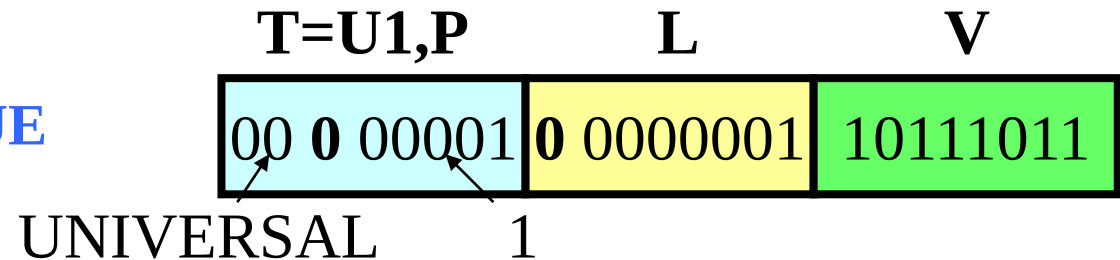
also legal!

sample-3 SEQUENCE OF INTEGER ::= {}

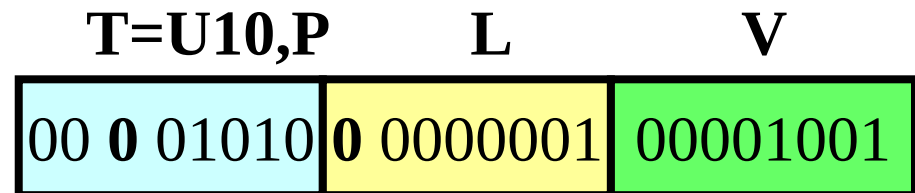
BER:**Encoding of tags****sample [3] EXPLICIT INTEGER ::= 0****sample [3] IMPLICIT INTEGER ::= 0**

BER:**NULL, BOOLEAN and ENUMERATED****null-type NULL ::= NULL**

-> have seen above,
just type = UNIVERSAL 5, L=0, no V

boolean BOOLEAN ::= TRUE

**sample-7 ENUMERATED { first (1),
second (-5), third (9) } ::= third**

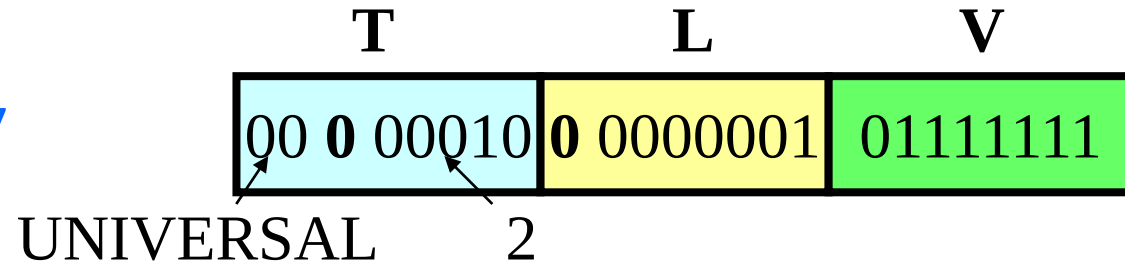


- sender's options:
 - for NULL type: none
 - for BOOLEAN: **short** or **long** forms, TRUE is any non-zero value
 - for ENUMARETED: **short** or **long** forms

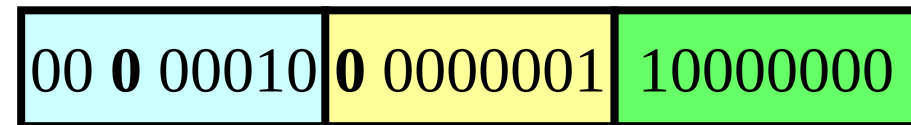
BER:

INTEGER

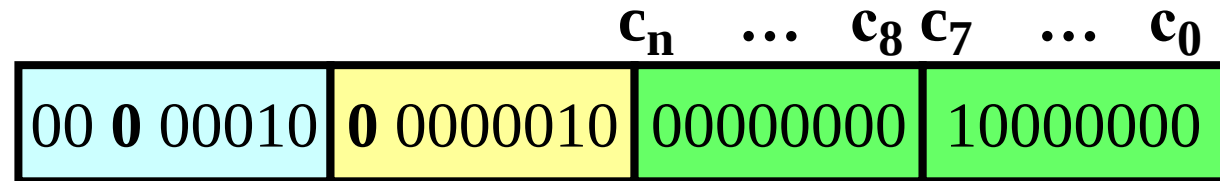
sample-4 INTEGER ::= 127



sample-5 INTEGER ::= -128



sample-6 INTEGER ::= 128



- first 9 leading bits of the V part shall not be all 0s or all 1s:
smallest possible V part
- sender's option: **short** or **long** form

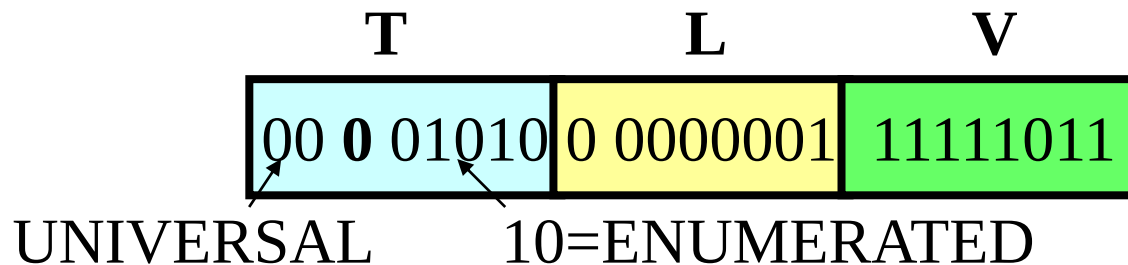
- coded as 2s complement: $\sum_{i=0}^{n-1} c_i 2^i - c_n 2^n$
- zero length V part is forbidden:
zero is coded with V=00H

BER:

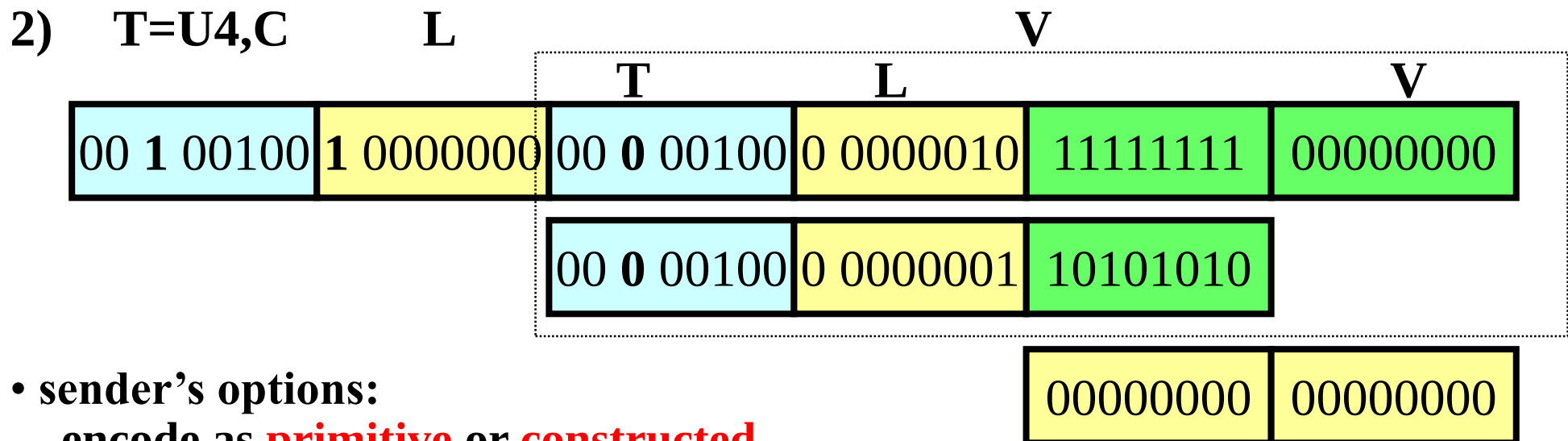
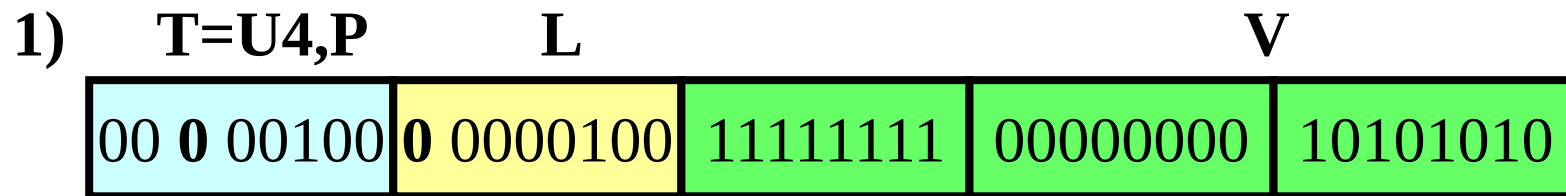
CHOICE

NO “OWN” ENCODING FOR CHOICE !

```
sample-8 CHOICE { flag    BOOLEAN,
                    value   ENUMERATED { first (1), second (-5) }
                    ::= value : second
```



- **Just the chosen inner type is encoded !**

BER:**OCTET STRING****sample-9 OCTET STRING ::= 'FF00AA'H**

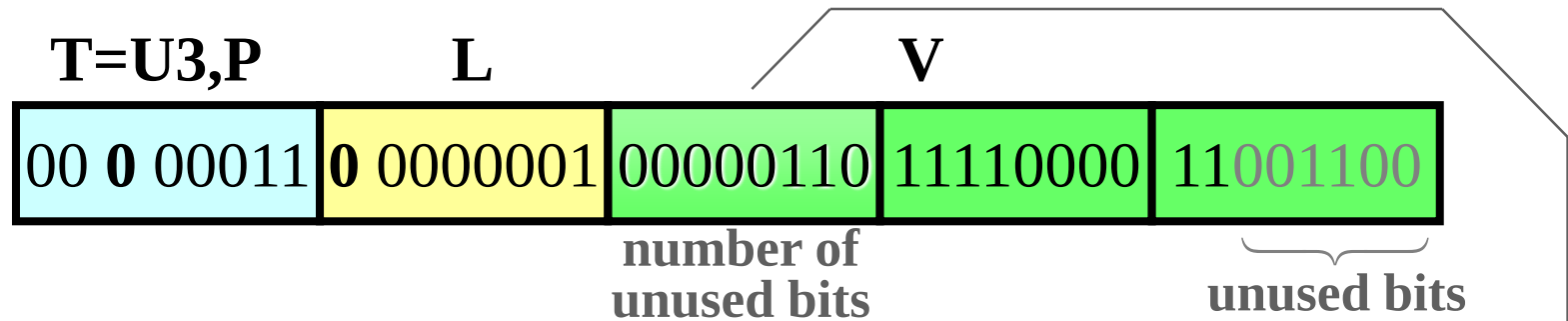
- sender's options:
 - encode as **primitive** or **constructed**
 - use the **short**, **long** or **indefinite** (only for constructed) length forms

BER:

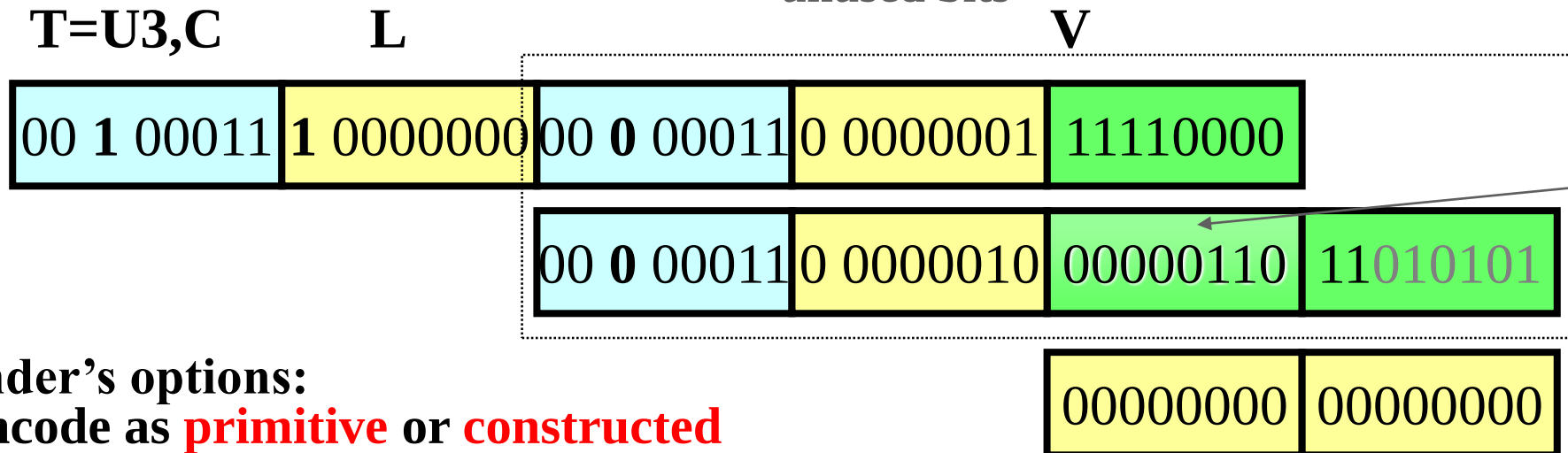
BIT STRING

sample-10 BIT STRING ::= '1111000011'B

1)



2)



• sender's options:

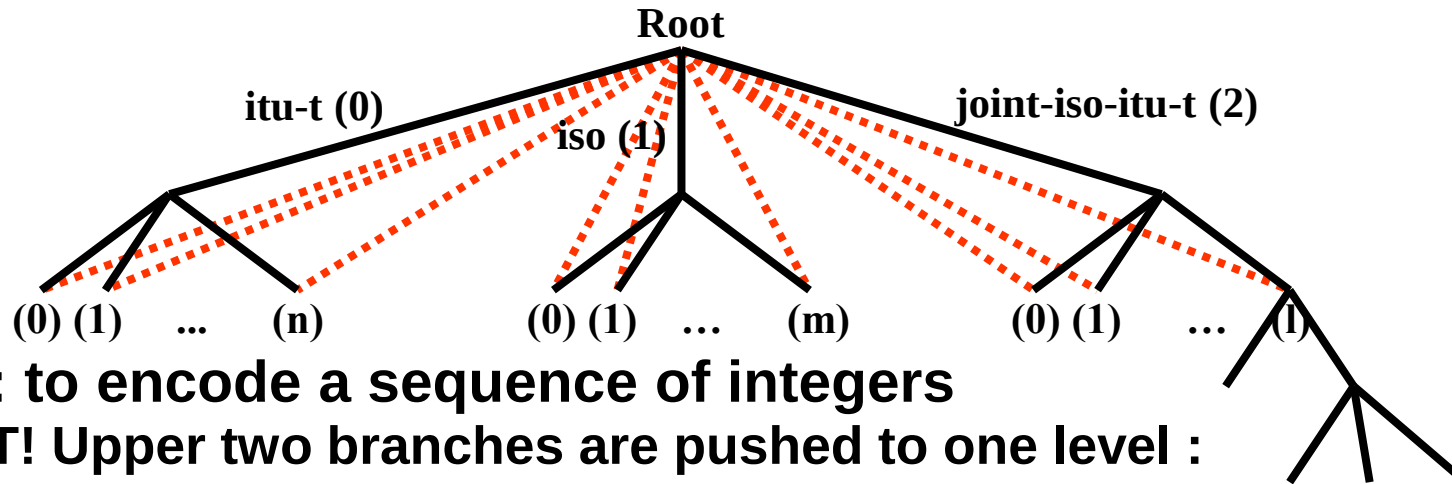
- encode as **primitive** or **constructed**
- use the **short**, **long** or **indefinite** (only for constructed) length forms
- remove trailing 0-s from a NamedBitList

BER:**SEQUENCE (OF), SET (OF)****sample-11 SEQUENCE OF INTEGER ::= { 3, 5 }****T=U16,C****L****V****sample-12 SET { flag1 BOOLEAN, flag2 [0] BOOLEAN }****T=U17,C****L****V****::= { flag2 TRUE, flag1 FALSE }**

**The principle is all the same, just the outer type
and encapsulated type(s) change**

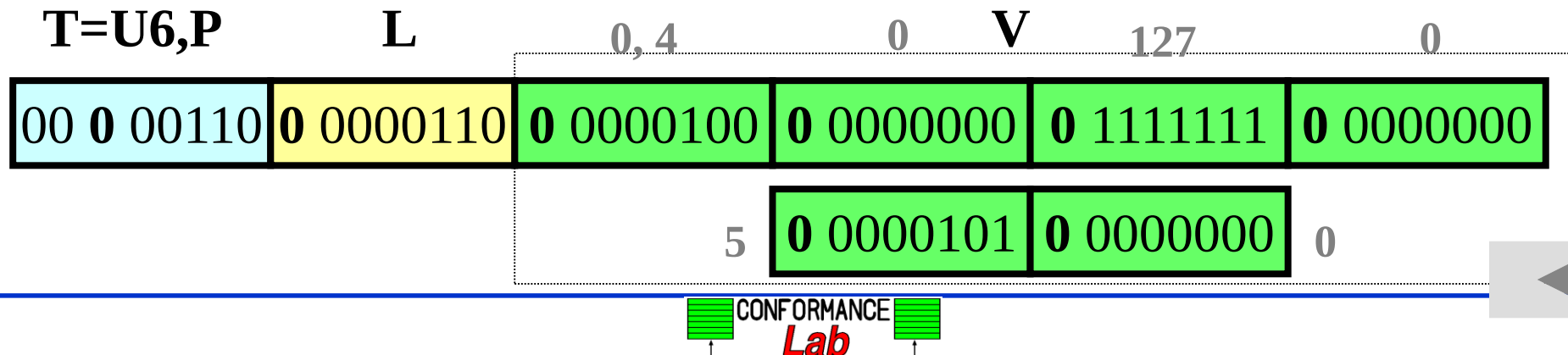
- encoding: always **constructed**, except when the V field is empty
- **short, long** or **indefinite** (except for primitive encoding) length forms



BER:**OBJECT IDENTIFIER**

- Task: to encode a sequence of integers
 - BUT! Upper two branches are pushed to one level :
40 * first-level-identifier + second-level-identifier
 - using MSB/octet as extension bit to identify the end of the integer

**ericsson-testing OBJECT IDENTIFIER ::= {itu-t(0) identified-organization(4)
etsi(0) reserved(127) etsi-identified-organization(0) ericsson(5) testing(0)}**



BER:***Ellipsis***

```
sample SEQUENCE { flag      BOOLEAN,
                    value     INTEGER (0..5),
                    which     SEQUENCE OF INTEGER OPTIONAL,
                    ... !PrintableString : "New parameters!",
                    new       [0] SEQUENCE {
                                flag1  BOOLEAN } OPTIONAL
} ::= { flag TRUE, value 3, which { 1, 2 }, new { flag1 FALSE } }
```

**No need for any special encoding: the TLV structure enables the decoder of an earlier implementation to sort out unexpected data using length fields of the outer type and added inner types.
Do not forget! All unknown elements can be identified due to the order in a SEQUENCE and/or distinct tags!**

CER and DER

Some restrictions in CER & DER

DER (Distinguished encoding rules)

CER (Canonical encoding rules)