

国际电信联盟

ITU-T

国际电信联盟
电信标准化部门

Z.146

(03/2006)

Z系列：电信系统使用的语言和一般性软件情况

正式描述技巧（FDT）— 测试和测试控制记法（TTCN）

**测试和测试控制记法第三版（TTCN-3）：通过TTCN-3
来使用ASN.1**

ITU-T Z.146建议书



ITU-T Z系列建议书
电信系统使用的语言和一般性软件情况

正式描述技巧 (FDT)	
规范和描述语言 (SDL)	Z.100-Z.109
正式描述技巧的应用	Z.110-Z.119
信息排序表 (MSC)	Z.120-Z.129
扩展的目标描述语言 (eODL)	Z.130-Z.139
测试和测试控制记法 (TTCN)	Z.140-Z.149
用户要求记法 (URN)	Z.150-Z.159
编程语言	
CHILL: ITU-T 高级语言	Z.200-Z.209
人机语言	
总则	Z.300-Z.309
基本句法和对话程序	Z.310-Z.319
用于视频显示终端的扩展 MML	Z.320-Z.329
人机接口规范	Z.330-Z.349
面向数据的人机接口	Z.350-Z.359
电信网络管理使用的人机接口	Z.360-Z.379
质量	
电信软件的质量	Z.400-Z.409
涉及协议的建议书中有质量的内容	Z.450-Z.459
方法	
认证与测试的方法	Z.500-Z.519
中间件	
分布或处理环境	Z.600-Z.609

欲了解更详细信息，请查阅 ITU-T 建议书目录。

ITU-T Z.146建议书

测试和测试控制记法第三版（TTCN-3）：通过TTCN-3来使用ASN.1

摘 要

本建议书定义了通过 TTCN-3（测试和测试控制记法第三版）（在 ITU-T Z.140 建议书中定义）来使用 ASN.1（在 ITU-T X.680 建议书、ITU-T X.681 建议书、ITU-T X.682 建议书、ITU-T X.683 建议书中定义）的方法。

本建议书的内容已从 ITU-T Z.140 建议书（04/2003）移入这个独立的建议书中。自此，进行了修正和编辑方面的改动，并都已包括在本建议书中。

来 源

ITU-T 第 17 研究组（2005-2008）按照 ITU-T A.8 建议书规定的程序，于 2006 年 3 月 16 日批准了 ITU-T Z.146 建议书。

前 言

国际电信联盟（ITU）是从事电信领域工作的联合国专门机构。ITU-T（国际电信联盟电信标准化部门）是国际电信联盟的常设机构，负责研究技术、操作和资费问题，并且为在世界范围内实现电信标准化，发表有关上述研究项目的建议书。

每四年一届的世界电信标准化全会（WTSA）确定 ITU-T 各研究组的研究课题，再由各研究组制定有关这些课题的建议书。

WTSA 第 1 号决议规定了批准建议书须遵循的程序。

属 ITU-T 研究范围的某些信息技术领域的必要标准，是与国际标准化组织（ISO）和国际电工技术委员会（IEC）合作制定的。

注

本建议书为简明扼要起见而使用的“主管部门”一词，既指电信主管部门，又指经认可的运营机构。

遵守本建议书的规定是以自愿为基础的，但建议书可能包含某些强制性条款（以确保例如互操作性或适用性等），只有满足所有强制性条款的规定，才能达到遵守建议书的目的。“应该”或“必须”等其它一些强制性用语及其否定形式被用于表达特定要求。使用此类用语不表示要求任何一方遵守本建议书。

知识产权

国际电联提请注意：本建议书的应用或实施可能涉及使用已申报的知识产权。国际电联对无论是其成员还是建议书制定程序之外的其它机构提出的有关已申报的知识产权的证据、有效性或适用性不表示意见。

至本建议书批准之日止，国际电联尚未收到实施本建议书可能需要的受专利保护的知识产权的通知。但需要提醒实施者注意的是，这可能并非最新信息，因此特大力提倡他们通过下列网址查询电信标准化局（TSB）的专利数据库：<http://www.itu.int/ITU-T/ipr/>。

© 国际电联 2006

版权所有。未经国际电联事先书面许可，不得以任何手段复制本出版物的任何部分。

目 录

	页码
1 范围.....	1
2 参考文献.....	1
3 定义和缩写.....	1
3.1 定义.....	1
3.2 缩写.....	1
4 引言.....	2
5 概述.....	2
6 对核心语言的修正.....	2
7 附加的 TTCN-3 类型.....	5
7.1 概述.....	5
7.2 附加的简单的基本类型和值.....	5
7.3 附加类型的子类型.....	6
8 ASN.1 和 TTCN-3 类型等价物.....	6
8.1 概述.....	6
8.2 标识符.....	7
9 ASN.1 数据类型和值.....	7
9.1 概述.....	7
9.2 ASN.1 标识符的范围.....	12
10 ASN.1 中的参数化.....	12
11 定义 ASN.1 消息模板.....	13
11.1 概述.....	13
11.2 ASN.1 使用 TTCN-3 模板语法来接收消息.....	13
11.3 模板字段排序.....	14
12 编码信息.....	14
12.1 概述.....	14
12.2 ASN.1 编码属性.....	14
12.3 ASN.1 变量属性.....	15
附件 A — 附加的 BNF 和静态语义.....	16
A.1 ASN.1 支持.....	16
附件 B — 预定义的 TTCN-3 函数.....	16
B.1 分解函数.....	16
附件 C — 预定义的对象标识符部件.....	17
参考资料.....	18

测试和测试控制记法第三版 (TTCN-3): 通过TTCN-3来使用ASN.1

1 范围

本建议书定义了一种通过 TTCN-3 来使用 ASN.1 (在 ITU-T X.680 建议书[2]、ITU-T X.681 建议书[3]、ITU-T X.682 建议书[4]、ITU-T X.683 建议书[5]中定义) 的规范方法。其他语言与 TTCN-3 之间的协调问题不在本建议书的讨论范围内。

2 参考文献

下列 ITU-T 建议书和其他参考文献的条款, 在本建议书中的引用而构成本建议书的条款。在出版时, 所指出的版本是有效的。所有的建议书和其他参考文献均会得到修订, 本建议书的使用者应查证是否有可能使用下列建议书或其他参考文献的最新版本。当前有效的 ITU-T 建议书清单定期出版。本建议书引用的文件自成一体时不具备建议书的地位。

- [1] ITU-T Recommendation Z.140 (2006), *Testing and Test Control Notation version 3 (TTCN-3): Core language.*
- [2] ITU-T Recommendation X.680 (2002), *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- [3] ITU-T Recommendation X.681 (2002), *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- [4] ITU-T Recommendation X.682 (2002), *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*
- [5] ITU-T Recommendation X.683 (2002), *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*
- [6] ITU-T Recommendation X.690 (2002), *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*
- [7] ITU-T Recommendation X.691 (2002), *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER).*
- [8] ITU-T Recommendation X.693 (2001), *Information technology – ASN.1 encoding rules: XML Encoding Rules (XER).*
- [9] Void.
- [10] ITU-T Recommendation T.100 (1988), *International information exchange for interactive Videotex.*
- [11] ITU-T Recommendation T.101 (1994), *International interworking for Videotex services.*
- [12] ITU-T Recommendation X.660 (2004), *Information technology – Open Systems Interconnection – Procedures for the operation of OSI registration authorities: General procedures and top arcs of the ASN.1 Object Identifier tree.*

3 定义和缩写

3.1 定义

就本建议书而言, ITU-T Z.140 建议书[1]中给出的术语和定义适用。

3.2 缩写

就本建议书而言, ITU-T Z.140 建议书[1]中给出的缩写以及下列缩写适用:

ASN.1 抽象语法记法1

4 引言

当通过 TTCN-3 来使用 ASN.1 时，在 ITU-T Z.140 建议书[1]第 4 节中给出的所有 TTCN-3 特征和语句仍将适用。另外，当支持本建议书时，TTCN-3 变得与 ASN.1 完全协调一致，通过 TTCN-3 模块，可以将之用作一种可选的数据类型和值语法。本建议书在 TTCN-3 模块中定义 ASN.1 的用法。用于结合 ASN.1 和 TTCN-3 的方法可以用于支持通过 TTCN-3 来使用其它类型和值系统。不过，本建议书对此未做详细定义。

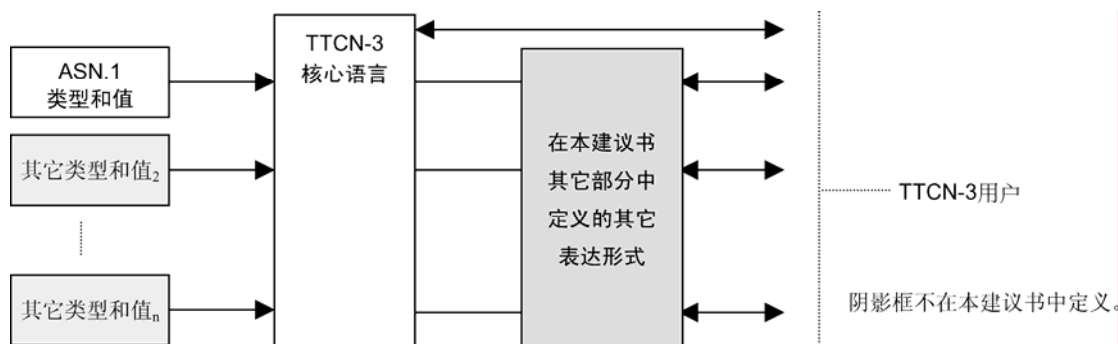


图 1/Z.146—核心语言和各种表达形式的用户观点

5 概述

TTCN-3 为在 TTCN-3 模块中使用 2002 版的 ASN.1 (在 ITU-T X.680 建议书[2]、ITU-T X.681 建议书[3]、ITU-T X.682 建议书[4]、ITU-T X.683 建议书[5]中定义) 提供了一个清晰的接口。目前未考虑 ASN.1 的 XML 相关性能。当从一个 ASN.1 模块引入时，将使用下列语言标识字符串中的其中一个：

- “ASN.1:2002” 针对2002版的ASN.1；
- “ASN.1:1997” 针对1997版的ASN.1；
- “ASN.1:1994” 针对1994版的ASN.1；
- “ASN.1:1988” 针对蓝皮书版的ASN.1。

注 1 — 语言标识符“ASN.1:1997”、“ASN.1:1994”和“ASN.1:1988”指的是基于所替代ITU-T建议书的ASN.1版本（包括基础文档和所有已出版的修正以及目前出版的、用于修正基础文档的技术勘误表）。如果通过TTCN-3来使用基于这些ASN.1版本的协议模块，那么在本建议书中包括它们的唯一目的是为了分配唯一标识符。当支持1997版ASN.1时，不考虑对ITU-T X.680建议书[2]修正3的支持。

注 2 — 当支持“ASN.1:1988”时，将依据ITU-T X.208建议书（蓝皮书）的语法和语义规则，从此类模块中引入ASN.1条目。

当通过 TTCN-3 来使用 ASN.1 时，11.18/X.680 [2]中所列的关键字不得用作 TTCN-3 模块中的标识符。ASN.1 关键字将遵从 ITU-T X.680 建议书[2]的要求。

6 对核心语言的修正

当在 TTCN-3 中使用 ASN.1 定义时，本条中给出的、对 TTCN-3 核心语言的修正（在 ITU-T Z.140 建议书[1]中定义）也将适用。下列条目编号指的是 ITU-T Z.140 建议书[1]中的条目编号。

条目3.1 定义

更改定义“已知类型”：利用以下之一：

known types (已知类型)：已定义类型、已输入的 ASN.1 类型和其它已输入的外部类型的集合。

在定义“根类型”之后插入以下注释：

注 — 在基于已输入ASN.1类型的类型情况下，根类型由相关的TTCN-3类型来确定（见第8条）。

条目3.2 缩写

增加以下所写：

ASN.1 抽象语法记法1

条目7.1 模块命名

添加至第一段第一句结尾处：

可选地，后跟一个对象标识符。

在注 1 后增加以下新的注释：

注 2 — 模块名称只能在对象标识符部分有所不同。不过，在这种情况下，应在入口处注意避免出现名称冲突，因为标识符前缀（见第7.5.8节）无法解决此类冲突。

条目7.5.0 概述

在第二段后新增以下段：

如果对象标识符作为输入语句中模块名称的一部分提供（自该模块引入定义），那么该对象标识符将用于确定恰当的模块。

条目7.5.8

在第一段后增加以下注释：

注 1 — 对象标识符值中用于解决名称冲突的规则在第6.1.0节条目d) 中给出。

条目14.3 模板匹配机制

将 **objid** 添加至第三段注释中的类型列表中：

注 — 可以省略以下类型：...

条目15.3 关系运算符

将 **objid** 加至第一段中的类型列表，对它，关系运算符是允许的（所有其它关系运算符都只能有 **integer** 类型（包括 **integer** 的衍生类型）、**float** 类型（包括 **float** 的衍生类型）、**objid** 类型...）。

在第二段后新增一个段：

如果两个 **objid** 值拥有相同数量的部件，并且所有位置上的数值都相同，那么它们是相等的。小于 (<)、大于 (>)、大于或等于 (>=) 以及小于或等于 (<=) 操作将使用 **objid** 值部件的数值来做决策，决策过程将遵循以下规则：

- 比较将从比较objid值的第一个部件开始；
- 如果第一个objid值中所比较部件的数值小于第一个objid值中对应部件的数值，那么第一个objid值小于第二个objid值；
- 如果两个比较部件的数值相等，那么以两个objid值的下一对部件继续进行比较；当中可以找到第一个带有较小数值部件的objid值小于另一个objid值；
- 如果两个objid值的所有比较部件都对相等，而一个objid值拥有更多的部件，另一个objid值没有一个未比较的部件，那么短的objid值小于长的objid值。

例如：

```
// 假定
const objid c_etsiMobNet := objid{itu_t identified_organization etsi(0)
                               mobile_domain(0) umts_Network(1)}
const objid c_etsiINNet  := objid{itu_t identified_organization etsi(0)
                               inDomain(1) in_Network(1)}
const objid c_etsiIN     := objid{itu_t identified_organization etsi(0)
                               inDomain(1)}
var objid v_etsiInIso    := objid{ iso identified_organization dod(6)
                               internet(1) private(4) enterprise(1) etsi(13019)}
// 那么
c_etsiMobNet == c_etsiINNet // 返回false
```

```

c_etsiMobNet < c_etsiINNet // 返回true, 原因是数字上mobile_domain(0)部件小于inDomain(1)部件。
c_etsiINNet == c_etsiIN // 返回false, 原因是c_etsiINNet拥有更多的部件。
c_etsiINNet > c_etsiIN // 返回true, 原因是c_etsiINNet拥有更多的部件。
v_etsiInIso <= c_etsiMobNet // 返回false, 原因是数字上itu_t(0)部件小于iso(1)部件。

```

条目16.1.3 预定义函数

将下列行加入表 10 的“其它函数”区中（TTCN-3 预定义函数的列表）。

其它函数		
	分解一个 objid 值	decomp

条目A.1.5 TTCN-3终端

将 `objid` 加入表 A.2（TTCN-3 特殊终端符号的列表）。

条目A.1.6.0 TTCN-3模块

在生成结果 `TTCN3ModuleId` 结尾处添加 “[DefinitiveIdentifier]”。

注 — 如果标准的其它部分没有任何其它变更，那么它变为：

```
'3. TTCN3ModuleId ::= ModuleIdentifier [DefinitiveIdentifier]'
```

条目A.1.6.1.8 入口定义

在生成结果 `GlobalModuleId` 结尾处添加 “[Dot ObjectIdentifierValue]”。

注 — 如果标准的其它部分没有任何其它变更，那么它变为：

```
'223. GlobalModuleId ::= ModuleIdentifier [Dot ObjectIdentifierValue]'
```

条目A.1.6.3 类型

在生成结果 `PredefinedType` 结尾处添加 “[ObjectIdentifierKeyword]”。

注 — 如果标准的其它部分没有任何其它变更，那么它变为：

```
'410. PredefinedType ::= BitStringKeyword |
                          BooleanKeyword |
                          CharStringKeyword |
                          UniversalCharString |
                          IntegerKeyword |
                          OctetStringKeyword |
                          HexStringKeyword |
                          VerdictTypeKeyword |
                          FloatKeyword |
                          AddressKeyword |
                          DefaultKeyword |
                          AnyTypeKeyword |
                          ObjectIdentifierKeyword'
```

条目A.1.6.4 值

在生成结果 `PredefinedValue` 结尾处添加 “[ObjectIdentifierValue]”。

注 — 如果标准的其它部分没有任何其它变更，那么它变为：

```
'434. PredefinedValue ::= BitStringValue |
                          BooleanValue |
                          CharStringValue |
                          IntegerValue |
                          OctetStringValue |
                          HexStringValue |
                          VerdictTypeValue |
                          EnumeratedValue |
                          FloatValue |
                          AddressValue |
                          OmitValue |
                          ObjectIdentifierValue'
```

条目C.14 结构化值中的元素数量

将 **objid** 添加至第一段第一句中的类型列表中（即“该函数返回模块参数、常量、变量或 **record** 的 **template**、**record of**、**set**、**set of** 或 **objid** 类型的实际元素数量...”）。

在第一段后新增以下段：

在 **objid** 值、模板或数组情况下，要返回的实际值为 **objid** 值中最后一个部件的序列号。

将以下新的例子加至例子段的结尾处：

```
// 假定
var objid v_etsiMobNet := objid{itu_t identified_organization etsi(0)
                               mobile_domain(0) umts_Network (1)}
// 那么
numElements := sizeof(v_etsiMobNet); // 返回5
```

7 附加的TTCN-3类型

7.1 概述

支持 ASN.1 用法的、附加的 TTCN-3 类型如表 1 所述。

表 1/Z.146—TTCN-3类型概述

类型类别	关键字	子类型
简单的基本类型	objid	列表

7.2 附加的简单的基本类型和值

为了支持在 TTCN-3 中使用 ASN.1，除了在核心语言建议书（ITU-T Z.140 建议书[1]）第 6.1 节中所规定的那些类型和值之外，还将支持以下简单的类型和值：

a) **objid**：一种类型，其不同值为以下值的集合：

- 符合附件A/X.660[12]要求的所有对象标识符值；以及
- ITU-T X.660建议书[12]中定义的值集之外的、所有语法上正确的对象标识符值（例如，不是在 ITU-T X.660建议书[12]中定义的、根之下的一个节点）。

除用下划线替代对象标识符中的连字号之外，**objid**类型的值记法都应符合第31条/X.680 [2]中给出的规则。

注 1 — 对象标识符部件的名称形式只能用于ITU-T X.660建议书[12]中所定义的部件。这些预定义的对象标识符部件在附件C中给出。在ITU-T X.660建议书[12]与附件C发生冲突的任何情况下，前者优先。

当对象标识符值记法中所引用的值标识符等同于任何预定义的部件名称时（即独立于记法内预定义部件或所引用值的位置），模块名称将作为所引用值名称的前缀，它在该模块中予以定义（见第7条/Z.140[1]中的TTCN-3模块定义）。将用点（.）来分隔前缀和标识符。预定义对象标识符部件名称还可以用名称“X660”作为前缀。

注 2 — 为了提高可读性，也建议在以下对象标识符值中使用“X660”前缀，即指的是与任何预定义部件名称有冲突的值标识符。

注 3 — 用于解决由入口引起之名称冲突的规则在第7.5.8节/Z.140[1]中进行定义。

例如：

```
objid{itu_t(0) identified_organization(4) etsi(0)}
// 或可选地
objid {itu_t identified_organization etsi(0)}
// 或可选地
objid { 0 4 0}

// 或可选地
const integer etsi := 0;
const objid itu_idOrg := objid{ itu_t identified_organization }
objid{ itu_idOrg etsi } // 注意，两个名称都引用值定义。

const integer x := 162;
objid{ itu_t recommendation x A.x } // 强制使用模块名称（“A”）来前缀模糊的标识符，
// 或者可选地，
objid{ itu_t recommendation X660.x A.x } // 即使使用“X660”前缀，也将出现模块名称。
```

7.3 附加类型的子类型

7.3.1 概述

用户定义的类型将用关键字 **type** 来表示。利用用户定义的类型，按照表 1，有可能创建类型的子类型（如列表）。

7.3.2 值列表

TTCN-3 允许规定 **objid** 类型不同值的一个列表。如 6.2.1/Z.140 [1]中所规定的那样，将应用子类型。

例如：

```
type objid MyListOfObjids (objid{0 4 0 0 1}, objid{0 4 0 1 1});
```

8 ASN.1 和 TTCN-3 类型等价物

8.1 概述

表 2 中所列 ASN.1 类型认为等同于其 TTCN-3 对应物。

表 2/Z.146—ASN.1 和 TTCN-3 等价物列表

ASN.1 类型	与TTCN-3等价物的映射
BOOLEAN	boolean
INTEGER	integer
REAL (Note)	float
OBJECT IDENTIFIER	objid
BIT STRING	bitstring
OCTET STRING	octetstring
SEQUENCE	record
SEQUENCE OF	record of
SET	set
SET OF	set of
ENUMERATED	enumerated
CHOICE	union
VisibleString	charstring
IA5String	charstring
UniversalString	universal charstring

注 — ASN.1 类型 REAL 等同于 TTCN-3 类型 **float**，直至基数是无限制的，或者显性地或隐性地限制为基数 10。ASN.1 记法允许显性限制，例如，通过内部子类型，但需来自 ASN.1-TTCN-3 类型映射观点，一个显性限制是一个 ASN.1 值记法。隐性限制可以通过对给定协议的文本描述来定义，即在 ASN.1 模块之外。不过，在两种情况下，都可以使用 TTCN-3 值记法，而不管基数是否在 ASN.1 中。

可以通过表 2 中给出的 TTCN-3 类型来使用的所有 TTCN-3 运算符、函数、匹配机制、值记法等，都可以通过对应的 ASN.1 类型来使用。

8.2 标识符

在将 ASN.1 标识符转换为 TTCN-3 标识符过程中，任何连字号“-”都将被转换为下划线“_”。

例如：

```
MyASN1module DEFINITIONS ::=
BEGIN
    Missleading-ASN1-Name ::=    INTEGER    -- ASN.1类型标识符使用“-”

END

module MyTTCNModule
{
    import from MyASN1module language "ASN.1:2002" all;

    const Missleading_ASN1_Name ExampleConst := 1;    // TTCN-3指的是使用下划线的ASN.1类型
}
```

9 ASN.1数据类型和值

9.1 概述

可以在 TTCN-3 模块中使用 ASN.1 类型和值。使用一个单独的 ASN.1 模块来定义 ASN.1 定义。通过其类型引用和值引用（依据 ASN.1 模块中的 9.3/X.680 和 9.4/X.680[2]来生成）来引用 ASN.1 类型和值。

例 1：

```
MyASN1module DEFINITIONS ::=
BEGIN
    Z ::=    INTEGER    -- 简单类型定义

    BMessage ::= SET    -- ASN.1类型定义
    {
        name    IA5String,
        title   VisibleString,
        date    IA5String
    }

    johnValues Bmessage ::=    -- ASN.1值定义
    {
        name    "John Doe",
        title   "Mr",
        date    "April 12th"
    }

    DefinedValuesForField1 Z ::= {0 | 1} -- ASN.1子类型定义
END
```

ASN.1 模块将符合 ITU-T X.680 建议书[2]、ITU-T X.681 建议书[3]、ITU-T X.682 建议书[4]、ITU-T X.683 建议书[5]的语法要求。一旦声明，就可以以类似使用其它 TTCN-3 模块中普通 TTCN-3 类型和值的方式，在 TTCN-3 模块中使用 ASN.1 类型和值(即引入要求的定义)。当在 TTCN-3 模块中引入 ASN.1 条目时，将为引入的每个 ASN.1 条目生成一个相关的类型或值。基于引入 ASN.1 条目的所有 TTCN-3 定义或赋值都将依据有关的相关类型或值作用的规则来生成。另外，当匹配常量、变量、模板或基于 ASN.1 声明的内嵌表达式时，匹配机制将使用相关类型。

通过应用以下转换规则，从 ASN.1 条目获得相关类型和值。转换开始于一个合法的 ASN.1 模块，结束于一个合法的 TTCN-3。次序对应单个转换的执行次序：

- 1) 忽略任何扩展标记和异常说明。
- 2) 忽略任何用户定义的限制（见第9节/X.682[4]）。
- 3) 忽略任何内容限制（见第9节/X.682[4]）。
- 4) 忽略任何样式限制（见第48.9节/X.680[2]）。
- 5) 通过用其代表的值集合替换包括的类型，为所有用所含之子类型进行约束的ASN.1类型，创建等价的TTCN-3子类型。有关将ASN.1类型约束转换为TTCN-3子类型的、更多的详细信息请参见表3。表3显示了ASN.1类型约束机制对不同ASN.1类型的适用性。对包含“否”的单元，对给定的类型不允许使用类型约束。带阴影的单元用于确定适用于某个给定类型的类型约束，单元中的文字用于定义在转换受限ASN.1类型时要用到的TTCN-3子类型机制。
- 6) 依据第24.4节/X.680[2]，对任何SEQUENCE类型执行COMPONENTS OF转换；依据第26.2节/X.680[2]，对任何包含关键字“COMPONENTS OF”的SET类型执行COMPONENTS OF转换。
- 7) 用其相关类型（通过将EMBEDDED PDV类型（见第33.5节/X.680[2]）相关类型中的内部子类型扩充为一个完全类型定义获得）替换EMBEDDED PDV类型。
- 8) 用其相关类型（通过将EXTERNAL类型（见第34.5节/X.680[2]）相关类型中的内部子类型扩充为一个完全类型定义获得）替换EXTERNAL类型（见注3）。
- 9) 用其相关类型（通过将CHARACTER STRING类型（见第40.5节/X.680[2]）相关类型中的内部子类型扩充为一个完全类型定义获得）替换CHARACTER STRING类型。
- 10) 用其相关类型（通过用其相关ASN.1类型替换INSTANCE OF DefinedObjectClass获得）替换INSTANCE OF类型（见第C.7节/X.681[3]），依据表2，用其TTCN-3等价物替换所有的ASN.1类型。结果类型为TTCN-3相关类型。
- 11) 忽略任何剩余的内部子类型（见注4）。
- 12) 忽略ASN.1类型中任何已命名的数和已命名的位。在ASN.1值中，用其值替换任何已命名的数，用不带尾0的比特串替换任何已命名的位或已命名的位的序列，用“1”替换由出现的名称所确定的位位置，用“0”替换其它的位位置。
- 13) 用选择类型所引用的类型替换任何选择类型；如果标明的选择类型（第29.1节/X.680[2]中的“类型”）是一个受限类型，那么必须选择所标明选择类型的父类型。
- 14) 将任何RELATIVE-OID类型或值转换为objid类型或值（见注5）。
- 15) 用其获得的相关类型替换任何以下严格的字符串类型（见注6）：
 - BMPString: **universal charstring** (char (0,0,0,0) .. char (0,0,255,255));
 - UTF8String: **universal charstring**;
 - NumericString: **charstring**, 限制为第37.2节/X.680[2]中给出的字符集;
 - PrintableString: **charstring**, 限制为第37.4节/X.680[2]中给出的字符集;
 - TeletexString 和 T61String: **universal charstring**, 限制为ITU-T T.61建议书中给出的字符集（见参考文献）;
 - VideotexString: **universal charstring**, 限制为ITU-T T.100建议书[10]和ITU-T T.101建议书[11]中给出的字符集;
 - GraphicString: **universal charstring**;
 - GeneralString: **universal charstring**。
- 16) 用**charstring**的类型或值替换任何GeneralizedTime 和 UTCTime的类型或值。
- 17) 用**universal charstring**的类型或值替换任何ObjectDescriptor的类型或值。
- 18) 用其引用的ASN.1条目（见注8）替换任何对象类别字段类型的记法（见第14节/X.681[3]）；出于转换目的（并且只能出于此目的），必须用元类型“OPEN TYPE”替换开放类型。

- 19) 用其引用的ASN.1条目替换所有来自对象记法的信息（见第15节/X.681[3]）。
- 20) 将表约束（见第10节/X.682[4]）回复给列表子类型，并忽略所有关系约束（见注7）。
- 21) 用以下相关TTCN-3类型替换所有出现的NULL类型：
- **type enumerated** <标识符> { NULL }，其中<标识符>是依据第7.2节转换的ASN.1类型引用。
- 22) 用元类型“开放类型”替换所有对开放类型的引用。
- 23) 依据表2，用其等价物替换ASN.1类型，用基于相关类型的等价TTCN-3值替换ASN.1值。（SET、SEQUENCE等）类型结构化ASN.1值中缺失的（即隐性地省略）可选字段，将在结果结构化TTCN-3值中被显性省略（见注9）。必须用**anytype**替换元类型“开放类型”。

注 1 — 单独的相关类型不能包含依据ASN.1类型对值进行正确编码所需的所有信息。处理系统提供正确编码所需附加信息的方法取决于实现方案，对用户是透明的；无需其知识来做出合法的TTCN-3声明或者涉及输入ASN.1类型和值的赋值。

注 2 — 当输入ENUMERATED类型时，也输入由用户指派给枚举类型的整数。

注 3 — 由编码方决定，EXTERNAL类型的数据值字段可以编码为单个ASN.1类型、八位字节联合类型或任何类型（见第8.18.1节/X.690 [6]）；如果用户想执行某种给定的编码形式或者在匹配时只想允许某种特定的编码形式，那么它将对类型或者给定的常量、变量、模板或模板字段使用适当的编码属性（见第11.3节）。

注 4 — 当依据受内部子类型限制的ASN.1类型来定义TTCN-3值或模板时，用户需要考虑到内部子类型。

注 5 — **objid**类型的等价物限制于只能用于值记法的语法。当对**objid**值（使用ASN.1编码规则得自ASN.1 RELATIVE-OID值）进行编码/解码时，依据为RELATIVE-OID类型规定的规则，进行编码/解码。

注 6 — VisibleString、IA5String 和 UniversalString 有其等价TTCN-3类型，直接进行替换。

注 7 — 当声明值和模板时，用户需要考虑到关系约束（也可通过工具隐性地进行处理）。

注 8 — 该替换不会影响应用于“有关对象类别字段类型的记法”自身的约束。

注 9 — (SET、SEQUENCE、EXTERNAL等)结构化ASN.1类型值中缺失的可选字段，等同于结构化TTCN-3值中被显性省略的字段。

例2:

```

module MyTTCNModule
{
    import from MyASN1module language "ASN.1:2002" all;

    const Bmessage MyTTCNConst:= johnValues;
    const DefinedValuesForField1 Value1:= 1;
}

```

注 10 — 不能直接从TTCN-3记法得到类型和值之外的ASN.1定义（即信息对象类别或信息对象集合）。在它们可以在TTCN-3模块中被引用之前，此类定义将被解析为ASN.1模板中的一个类型或值。

表 3/Z.146—对TTCN-3子类型转换的ASN.1类型约束

类型（或者通过标记或子类型源自这样一个类型）	单个值	包含的子类型 ^{b)}	值范围	大小约束	允许的字母表	类型约束	内部的子类型 ⁱ⁾	样式约束	用户定义的约束	表约束 ^{k)}	关系约束 ^{k)}	内容约束
比特串	列表	单个值: 列表, 大小: 长度	否	长度	否	否	否	否	忽略	否	否	忽略
布尔	列表	列表	否	否	否	否	否	否	忽略	否	否	否
选择	列表	列表	否	否	否	否	转换为完全类型	否	忽略	否	否	否
嵌入的 pdv ^{a)}	列表	否	否	否	否	否	转换为完全类型	否	忽略	否	否	否
枚举的	列表	列表	否	否	否	否	否	否	忽略	否	否	否
外部的 ^{a)}	列表	否	否	否	否	否	转换为完全类型	否	忽略	否	否	否
^{a), b)} 的实例	列表	列表	否	否	否	否	转换为完全类型	否	忽略	否	否	否
整数	列表	单个值: 列表, 值范围: 范围	范围	否	否	否	否	否	忽略	否	否	否
空	忽略	忽略	否	否	否	否	否	否	忽略	否	否	否
对象类别字段类型	^{c)}	^{c)}	否	否	否	否	否	否	忽略	列表	忽略	否
对象描述符 ^{e)}	列表	单个值: 列表, 大小: 长度, perm.alphabet: 范围	否	长度	范围	否	否	否	忽略	否	否	否
对象标识符	列表	列表	否	否	否	否	否	否	忽略	否	否	否
八位字节串	列表	单个值: 列表, 大小: 长度	否	长度	否	否	否	否	忽略	否	否	忽略
开放类型 ^{f)}	否	否	否	否	否	带列表约束的 anytype	否	否	忽略	否 ^{m)}	否 ^{m)}	否
实数	列表	单个值: 列表, 值范围: 范围	范围	否	否	否	转换为完全类型	否	忽略	否	否	否
相关对象标识符 ^{d)}	列表	列表	否	否	否	否	否	否	忽略	否	否	否

表 3/Z.146—对TTCN-3子类型转换的ASN.1类型约束

类型（或者通过标记或子类型源自这样一个类型）	单个值	包含的子类型 ^{b)}	值范围	大小约束	允许的字母表	类型约束	内部的子类型 ⁱ⁾	样式约束	用户定义的约束	表约束 ^{k)}	关系约束 ^{k)}	内容约束
受限的字符串类型	列表	单个值：列表，大小：长度，perm.alphab.: 范围	范围	长度	范围	否	否	忽略 ^{g)}	忽略	否	否	否
序列	列表	list	否	否	否	否	转换为完全类型	否	忽略	否	否	否
.....的序列	列表	单个值：列表，值范围：范围	否	长度	否	否	转换为完全类型	否	忽略	否	否	否
集合	列表	列表	否	否	否	否	转换为完全类型	否	忽略	否	否	否
.....的集合	列表	单个值：列表，值范围：范围	否	长度	否	否	转换为完全类型	否	忽略	否	否	否
时间类型 ^{a)}	列表	列表	否	否	否	否	否	否	忽略	否	否	否
非受限的字符串类型 ^{a)}	列表	否	否	长度（适用于字段“串值”）	否	否	转换为完全类型	否	忽略	否	否	否

- a) 从 TTCN-3 角度来看，这些类型等同于其相关类型。
- b) Instance of 相关类型的 Type-id 字段将由&id 字段的类型替代，值字段为 anytype（附件 C/X.681 [3]）。
- c) 由引用的类型替代，因此适用于引用的类型。
- d) 看作是来自 TTCN-3 的对象标识符。
- e) 其相关类型为受限的字符串类型。
- f) 开放类型由 anytype 替代。
- g) 字符样式只能用在 TTCN-3 中的常量、变量、模板和模块参数中，但不能用于子类型。
- h) 包含的子类型约束将由入口处的文字约束替代。
- i) 本栏中的信息与 ASN.1 术语的 TTCN-3 观点相关。将依据根类型进行编码/解码；因此，还必须保存未显示在本表中的、用于编码的额外信息。
- k) 只适用于有关对象类别字段类型的记法。
- m) 当使用有关对象类别字段类型的记法来定义开放类型时，它适用（见上面所述）。

9.2 ASN.1标识符的范围

引入的 ASN.1 标识符遵循与引入的 TTCN-3 类型和价值相同的范围规则（见 5.3/Z.140[1]）。

10 ASN.1中的参数化

允许通过 TTCN-3 模块来引用参数化的 ASN.1 类型和价值定义。不过，对在 TTCN-3 模块中使用的所有 ASN.1 参数化定义，都应提供实际参数（不允许是开放类型），提供的实际参数将在编译时予以解析。

TTCN-3 核心语言不支持 ASN.1 条目的入口，它唯一地使用 ASN.1 特定的对象作为形参或实参。因此在 TTCN-3 中使用之前，将在 ASN.1 部分中对涉及对象（不能直接在 TTCN-3 核心语言中进行定义）的、ASN.1 特定的参数化进行解析。ASN.1 特定的对象为：

- a) 信息对象类别；
- b) 信息对象；
- c) 信息对象集合。

例如，下面的内容不是合法的，原因是它定义的 TTCN-3 类型将一个 ASN.1 对象集合作为一个实际的参数。

```
MyASN1module DEFINITIONS ::=
BEGIN
  -- ASN.1 Module definition

  -- Information object class definition
  MESSAGE ::= CLASS { &msgTypeValue  INTEGER UNIQUE,
                     &MsgFields}

  -- Information object definition
  setupMessage MESSAGE ::= { &msgTypeValue  1,
                             &MsgFields    OCTET STRING}

  setupAckMessage MESSAGE ::= { &msgTypeValue  2,
                               &MsgFields    BOOLEAN}

  -- Information object set definition
  MyProtocol MESSAGE ::= { setupMessage | setupAckMessage}

  -- ASN.1 type constrained by object set
  MyMessage{ MESSAGE : MsgSet} ::= SEQUENCE
  {
    code      MESSAGE.&msgTypeValue({ MsgSet}),
    Type      MESSAGE.&MsgFields({ MsgSet})
  }
END

module MyTTCNModule
{
  // TTCN-3模块定义
  import from MyASN1module language "ASN.1:2002" all;

  // 非法的TTCN-3类型，它将对象集合作为了参数。
  type record Q(MESSAGE MyMsgSet) ::= { Z          field1,
                                         MyMessage(MyMsgSet) field2}
}
```

为使之成为一个合法的定义，必须定义额外的 ASN.1 类型 My Message1，如下所示。这解决了信息对象集合参数化问题，并因此可以直接在 TTCN-3 模块中使用。

```
MyASN1module DEFINITIONS ::=
BEGIN
  -- ASN.1模块定义

  ...

  MyProtocol MESSAGE ::= { setupMessage | setupAckMessage}
```

```

-- 额外的ASN.1类型，用于解决对象集合参数化问题。
MyMessage1 ::= MyMessage{ MyProtocol}
END

module MyTTCNModule
{
// TTCN-3模块定义
import from MyASN1module language "ASN.1:2002" all;

// 合法的TTCN-3类型，没有对象集合作为参数。
type record Q := { Z field1,
                  MyMessage1 field2}
}

```

11 定义ASN.1消息模板

11.1 概述

如果在 ASN.1 中定义消息，例如，使用 SEQUENCE（或者可能使用 SET），那么对 **send** 和 **receive** 事件，可以使用 ASN.1 值语法来规定实际的消息。

例如：

```

MyASN1module DEFINITIONS ::=
BEGIN
-- ASN.1模块定义

-- 消息定义
MyMessageType ::= SEQUENCE
{
field1 [1] IA5STRING, // 像TTCN-3字符串
field2 [2] INTEGER OPTIONAL, // 像TTCN-3整数
field3 [4] Field3Type, // 像TTCN-3记录
field4 [5] Field4Type // 像TTCN-3数组
}

Field3Type ::= SEQUENCE {field31 BIT STRING, field32 INTEGER, field33 OCTET STRING},
Field4Type ::= SEQUENCE OF BOOLEAN

-- 可以有以下值:
myValue MyMessageType ::=
{
field1 "A string",
field2 123,
field3 {field31 '11011'B, field32 456789, field33 'FF'O},
field4 {true, false}
}
END

```

11.2 ASN.1使用TTCN-3模板语法来接收消息

在标准的 ASN.1 语法中，不支持匹配机制。因此，如果希望通过 ASN.1 接收消息来使用匹配机制，那么将替代使用有关接收模板的 TTCN-3 语法。注意，该语法包括部件引用，以便能够引用 ASN.1 SEQUENCE、SET 等中的单个部件。

例如：

```

import from MyASN1module language "ASN.1:2002" {
type myMessageType
}

// 使用TTCN-3内匹配机制的消息模板可以是:
template myMessageType MyValue :=
{
field1 := "A"<?>"tr"<*>"g",
field2 := *,
field3.field31 := '110??'B,
field3.field32 := ?,
field3.field33 := 'F?'O,
field4.[0] := true,
field4.[1] := false
}

```

```
// 同样，下列语法是合法的。
template myMessageType MyValue:=
{
    field1 := "A"<?>"tr"<*>"g",           // 带通配符的串。
    field2 := *,                           // 任何整数或者根本没有整数。
    field3 := {'110??'B, ?, 'F?'O},
    field4 := {?, false}
}
```

11.3 模板字段排序

当 TTCN-3 模板用于 ASN.1 类型时，模板中字段次序的意义将取决于用于定义消息类型的 ASN.1 构件的类型。例如，如果使用 SEQUENCE 或 SEQUENCE OF，那么将以模板中规定的次序来发送或匹配消息字段。如果使用 SET 或 SET OF，那么可以以任何次序来发送或匹配消息字段。

12 编码信息

12.1 概述

TTCN-3 允许引用编码规则内的编码规则和变更，以便实现与各种 TTCN-3 语言元素的关联。也可能定义无效的编码。依据以下语法，使用 **with** 语句来规定该编码信息。

```
例如：
module MyModule
{
    :
    import from MyASN1module language "ASN.1:2002" {
        type myMessageType
    }
    with {
        encode "PER-BASIC-ALIGNED:1997" // MyMessageType的所有实例都应使用PER:1997来编码。
    }
    :
} // 模块结束
with { encode "BER:1997" } // 整个模块（测试组）的缺省编码为BER:1997。
```

12.2 ASN.1编码属性

以下各串为 ASN.1 当前版本的预定义（标准）编码属性：

- a) “BER:2002” 意味着依据ITU-T X.690建议书[6]（BER）进行编码；
- b) “CER:2002” 意味着依据ITU-T X.690建议书[6]（CER）进行编码；
- c) “DER:2002” 意味着依据ITU-T X.690建议书[6]（DER）进行编码。
- d) “PER-BASIC-UNALIGNED:2002” 意味着依据ITU-T X.691建议书[7]（非联合PER）进行编码；
- e) “PER-BASIC-ALIGNED:2002” 意味着依据ITU-T X.691建议书[7]（联合PER）进行编码；
- f) “PER-CANONICAL-UNALIGNED:2002” 意味着依据ITU-T X.691建议书[7]（规范的非联合PER）进行编码；
- g) “PER-CANONICAL-ALIGNED:2002” 意味着依据ITU-T X.691建议书[7]（规范的联合PER）进行编码；
- h) “BASIC-XER:2002” 意味着依据ITU-T X.693建议书[8]（基本的XML编码规则）进行编码；
- i) “CANONICAL-XER:2002” 意味着依据ITU-T X.693建议书[8]（规范的XML编码规则）进行编码；
- j) “EXTENDED-XER:2002” 意味着依据ITU-T X.693建议书/修正1[8]（扩展的联合XML）进行编码。

也可以使用之前 ASN.1 版本规则的编码（如 1988、1994 或 1997）。在这种情况下，必须对日期做相应替换。例如，对 ASN.1 1997，应用以下编码属性：“BER:1997”、“CER:1997”、“DER:1997”、“PER-BASIC-UNALIGNED:1997”、“PER-BASIC-ALIGNED:1997”、“PER-CANONICAL-UNALIGNED:1997”和“PER-CANONICAL-ALIGNED:1997”。

12.3 ASN.1变量属性

下列串为预定义的（标准的）变量属性。只有当与预定义的 ASN.1 编码属性联合使用时，它们才具有预定义的含义（见第 11.2 节）。当与其它属性联合使用时，或者当用于不带属性的 TTCN-3 对象时，这些预定义属性的处理超出了本建议书的讨论范围（见注 1）。

- a) “长度形式1”指的是：在BER、CER和DER编码情况下（第10.9节/X.691 [7]），或者在任何PER编码形式下，由单个八位字节长度决定时（见注2），给出的值将只能用长度八位字节的短形式来编码和解码（见第8.1.3节/X.690[6]）。
- b) “长度形式2”指的是：在BER、CER和DER编码情况下（第10.9节/X.691 [7]），或者在任何PER编码形式下，由两个八位字节长度决定时（见注2），给出的值将只能用长度八位字节的长形式来编码和解码（见第8.1.3节/X.690[6]）。
- c) “长度形式3”指的是：在BER、CER和DER编码情况下，给出的值将只能用长度八位字节的不确定形式来编码和解码（见第8.1.3节/X.690[6]）。
- d) “实数基数2”指的是：将依据实数二进制编码形式来对给出的值进行编码和解码。当用于任何分组类型时（例如，用于组或整个入口语句），该属性只能用于常量、变量或模板，它将只对这些TTCN-3对象有影响。
- e) “单个ASN 1类型”、“八位字节对齐的”、“任意的”指的是：只有当它已由发送方使用规定的选项做了编码时（见第8.18节/X.690[6]），才能使用由变量属性规定的“编码”选项或者在接收时匹配的“编码”选项来对值的“编码”字段（它将基于一个ASN.1 EXTERNAL类型）进行编码。该属性只能通过引入的ASN.1 EXTERNAL类型以及基于这种类型的常量、变量、模板或模板字段来使用。当由任何分组类型使用时（例如，TTCN-3组、引入某种定义、引入所有语句等），它将只对基于ASN.1 EXTERNAL类型的TTCN-3对象有影响。如果不满足第8.18.6节到第8.18.8节/X.690[6]中的条件集和规定的属性，那么这将引起一个运行时错误。
- f) “TeletexString”指的是：对给出的值，将作为ASN.1类型TeletexString来编码和解码（见第8.20节/X.690 [6]和第26节/X.691 [7]）。
- g) “VideotexString”指的是：对给出的值，将作为ASN.1类型VideotexString来编码和解码（见第8.20节/X.690 [6]和第26节/X.691 [7]）。
- h) “GraphicString”指的是：对给出的值，将作为ASN.1类型GraphicString来编码和解码（见第8.20节/X.690 [6]和第26节/X.691 [7]）。
- i) “GeneralString”指的是：对给出的值，将作为ASN.1类型GeneralString来编码和解码（见第8.20节/X.690 [6]和第26节/X.691 [7]）。

注 1 — 这些属性可以以不同于当前条款中规定的含义，重用于实现方案特定的编码规则中，可以忽略之，或者产生一个告警/错误提示。不过，所用的策略将取决于实现方案。

注 2 — 应用这些变量属性可能导致无效的ASN.1编码（例如，对BER中的原始值，使用不确定的长度形式，或者不使用所需的最小长度八位字节数量）。有意允许这样，用户将慎重地把这些变量属性分配给用于接收的常量、变量、模板或模板字段。

附件 A

附加的BNF和静态语义

为了支持在 TTCN-3 中使用 ASN.1, 将通过本附件中所规定的 BNF 和语义规则来补充附件 A/Z.140[1]中的 TTCN-3 语法。

A.1 ASN.1支持

```
xxx. DefinitiveIdentifier ::= Dot ObjectIdentifierKeyword "{" DefinitiveObjIdComponentList "}"
xxx. ObjectIdentifierKeyword ::= "objid"
xxx. DefinitiveObjIdComponentList ::= {DefinitiveObjIdComponent}+
xxx. DefinitiveObjIdComponent ::= NameForm |
                                   DefinitiveNumberForm |
                                   DefinitiveNameAndNumberForm
xxx. DefinitiveNumberForm ::= Number
xxx. DefinitiveNameAndNumberForm ::= Identifier "(" DefinitiveNumberForm ")"
xxx. ObjectIdentifierValue ::= ObjectIdentifierKeyword "{" ObjIdComponentList "}"
xxx. ObjIdComponentList ::= {ObjIdComponent}+
xxx. ObjIdComponent ::= NameForm |
                        NumberForm |
                        NameAndNumberForm |
                        ReferencedValue
/* 静态语义-ReferencedValue是一个对象标识符值。 */
xxx. NameForm ::= Identifier
xxx. NumberForm ::= Number | ReferencedValue
/* 静态语义-ReferencedValue是一个非负的整数值。 */
xxx. NameAndNumberForm ::= Identifier "(" NumberForm ")"
```

附件 B

预定义的TTCN-3函数

B.1 分解函数

```
decomp (in objid inval, in integer index, in integer count) return objid
```

该函数返回一个对象标识符值, 包含输入对象标识符值 (inval) 的一个片段 (部件序列)。片段的起始点由参数 (下标) 的第二个来定义。“下标”参数的实际值是一个非负的整数。下标从 0 开始计, 用于确定输入对象标识符值的第一个部件。第三个输入参数 (计数) 定义返回 objid 值中的部件数量。“计数”参数的实际值是一个正的非零整数。“下标”和“计数”参数的总和小于或等于输入 objid 值中部件数量减 1。

例如:

```
var objid v_etsiMobNet := objid{itu_t identified_organization etsi(0)
                               mobile_domain(0) umts_Network (1)}

decomp (v_etsiMobNet, 0, 2) // 返回 {itu_t identified_organization}

decomp (v_etsiMobNet, 2, 3) // 返回 {etsi(0) mobile_domain(0) umts_Network (1)}

decomp (v_etsiMobNet, 0, 0) // 引起错误, 原因是返回的部件数量将大于0。

decomp (v_etsiMobNet, 0, 6) // 引起错误, 原因是objid输入值包含的部件数量小于6。
```

附 件 C

预定义的对象标识符部件

ITU-T X.660 建议书[12]定义了对象标识符部件树，如下所示。只有在 ITU-T X.660 建议书[12]中定义的对象标识符部件将使用对象标识符值记法中的名称形式（不定义部件的数值）。只有当在其预定义位置上使用时，这些预定义部件才具有规定的数值。出于历史原因，保留斜体形式的名称；因此，不赞成在 TTCN-3 代码中使用它们，但建议 TTCN-3 工具应该能够识别它们，并用正确的数值替代它们。

注 — 依

据TTCN-3语法给出以下名称，也就是说，所有的破折号均由下划线替代。

```
itu_t(0), ccitt(0), itu_r(0)
  recommendation(0)
    a(1)
    d(4)
    e(5)
    f(6)
    g(7)
    h(8)
    i(9)
    j(10)
    k(11)
    l(12)
    m(13)
    n(14)
    o(15)
    p(16)
    q(17)
    r(18)
    s(19)
    t(20)
    u(21)
    v(22)
    x(24)
    y(25)
    z(26)
  question(1)
  administration(2)
  network_operator(3)
  identified_organization(4)
  r_recommendation(5)
iso(1)
  standard(0)
  registration_authority(1)
  member_body(2)
  identified_organization(3)
joint_iso_itu_t(2), joint_iso_ccitt(2)
```

参考资料

- ISO/IEC 6429:1992, *Information technology – Control functions for coded character sets*.
- ITU-T Recommendation T.50 (1992), *International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange*.
- ITU-T Recommendation X.208, *Specification of Abstract Syntax Notation One (ASN.1)*.
- ISO/IEC 8859-1:1998, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*.
- ITU-T Recommendation T.61 (1993), *Character repertoire and coded character sets for the international teletex service*.
- 可以在以下网址免费得到一个对象标识符（OID）知识库：<http://oid.elibel.tm.fr>。

注 — 以上网址的内容计划移至ITU-T网站中。

ITU-T 系列建议书

A系列	ITU-T工作的组织
D系列	一般资费原则
E系列	综合网络运行、电话业务、业务运行和人为因素
F系列	非话电信业务
G系列	传输系统和媒质、数字系统和网络
H系列	视听及多媒体系统
I系列	综合业务数字网
J系列	有线网络和电视、声音节目及其它多媒体信号的传输
K系列	干扰的防护
L系列	电缆和外部设备其它组件的结构、安装和保护
M系列	电信管理，包括TMN和网络维护
N系列	维护：国际声音节目和电视传输电路
O系列	测量设备的技术规范
P系列	电话传输质量、电话设施及本地线路网络
Q系列	交换和信令
R系列	电报传输
S系列	电报业务终端设备
T系列	远程信息处理业务的终端设备
U系列	电报交换
V系列	电话网上的数据通信
X系列	数据网、开放系统通信和安全性
Y系列	全球信息基础设施、互联网协议问题和下一代网络
Z系列	用于电信系统的语言和一般软件问题