第 4 章　ASN.1 及其编码规则

4.1　ASN.1

ASN.1 是 Abstract Syntax Notation One（抽象文法描述语言）的缩写。ASN.1 对于大多数人来讲似乎很陌生，然而事实上它就应用在我们生活的周围。目前北美、欧洲和日本等地使用的移动电话，都是基于 TCAP 消息协议的。TCAP 消息协议中的消息是采用 ASN.1 描述的，使用了 BER（Basic Encoding Rules）编码规则，共同实现了移动电话的呼叫。可以说，在一部移动电话与另外一部电话之间通信时，是 ASN.1 协助实现了两部话机之间的呼叫。后来制定的有关地对空和地对地等通信协议都是使用 ASN.1 描述的，并采用 PER（Packed Encoding Rules）编码规则。除此之外，ASN.1 和编码规则还被联邦快递用于大量地传输信息；还有许多大公司如 HP、IBM、SUN 等，使用 ASN.1 描述其打印机打印作业管理的标准接口。有名的简单网络管理协议（Simple Network Management Protocol，SNMP）就是用 ASN.1 对所有数据进行描述的。

ASN.1 是一种对分布计算机系统之间交换的数据消息进行抽象描述的规范化语言。以前，ASN.1 只用于撰写国际通用标准。然而，随着 ASN.1 软件工具的出现，ASN.1 已经用于生成应用程序编程语言代码，成为各种消息系统应用的核心。现在 ASN.1 成为描述通信协议的标准文法，而且对通信协议的描述，不用再区分通信程序实现的编程语言和通信数据的原始表示，也不用再区分应用系统的复杂或简单。

总之，ASN.1 是一种国际标准，它为抽象数据结构的描述说明定义了一种记法。ASN.1 使用抽象法对各种编程语言定义的数据类型进行了重新定义，将所有数据分为两大类：基本类型（如整型、布尔类型、字符串类型和比特串类型等）和结构类型（如结构、链表和选择类型等）。使用抽象法描述的系统让设计者可以只关心系统的某部分，而不必关心系统中的某一部分功能如何实现或者所代表的内容。抽象法作为软件开发管理的关键，已经为越来越多的软件设计开发人员所认可。对于某个开放使用的应用层程序，抽象法可以简化程序的说明，同时可以将程序的执行过程及其相关部分表述成抽象语言；当上一级程序使用下一级应用程序时也使用抽象法简化其过程，抽象法在越来越多的软件中得到应用。简单地说，抽象法通过抽象文法描述将数据结构进行抽象化描述，使用抽象文法描述的应用消息作为该应用的消息协议，如果该应用涉及多级应用程序，每一级应用只构造或解释与该级应用相关的信息，而不必关心上下级应用的信息。

ASN.1 作为抽象描述文法，将现有的数据类型抽象描述成近 20 种数据类型。这些数据类型主要分为两大类：基本类型和结构类型。

基本类型又称为原子类型，是构成其他结构类型的成员类型，主要包括：

① 布尔类型：BOOLEAN

② 整型：INTEGER

③ 比特串：BIT STRING

④ 字节串：OCTET STRING
⑤ 空：NULL
⑥ 对象标识：OBJECT IDENTIFIER
⑦ 可打印字符串：PrintableString
⑧ IA5 字符串：IA5String
⑨ 可见字符串：VisibleString
⑩ 数字字符串：NumericString
⑪ BMP 字符串：BMPString
⑫ 枚举类型：Enumerated
⑬ UTC 时间类型：UTCTime
⑭ Generalized 时间类型：GeneralizedTime
⑮ 任意类型：ANY

其中，布尔类型是任意取值只为 0 或者 1 的数据类型；整型是任意的整数数据类型；比特串是以比特为单位任意的字符串（0，1 串）；字节串是以字节为单位的任意字符串；空类型是 NULL；对象标识是使用一串数字标识一个实体，例如一种算法或一种属性；可打印字符串是任意可打印字符组成的字符串；IA5 字符串是任意 ASCII 字符组成的字符串；数字字符串是字符 0 到 9 任意组成的字符串；BMP 字符串是使用两个字节表示一个字节数据的字符串；枚举类型与编程语言中描述的枚举类型一样；UTC 时间类型是格林尼治时间的数据类型；Generalized 时间类型是表示本地时间的数据类型；任意类型是对使用编码规则编码生成的信息定义的数据类型。

结构类型又称为复合类型，主要包含

① 有序成员固定结构：SEQUENCE
② 无序成员固定结构：SET
③ 有序成员待定结构：SEQUENCE OF
④ 无序成员待定结构：SET OF
⑤ 选择类型：CHOICE

其中，有序成员固定结构是指使用前已确定数据成员的个数和顺序的结构体类型；无序成员固定结构是指使用前已确定数据成员的个数，但未确定数据成员顺序的结构体类型；有序成员待定结构是指使用前已确定数据成员数据，使用时才确定数据成员的个数的结构体链表类型；无序成员待定结构是指使用时才确定数据成员的个数和顺序的结构体链表类型；选择类型是由几种数据类型的数据成员构成的共同体类型。

通过 ASN.1 抽象定义后的数据类型几乎概括了现实世界中存在的所有数据类型，具有相当的通用性。在制定应用系统消息协议时就使用这些数据类型描述消息结构，屏蔽了各种编程语言自身的数据类型，提高了消息通用性。同时，由于使用了这些抽象数据类型描述消息协议，还克服了原编程语言描述消息结构的许多弊病。例如，在使用编程语言描述的消息协议中，协议设计人员为了使协议具有一定的扩展性，需要在一些数据结构中保留字段，由于定义保留字段时无法确定以后要扩展的数据类型，在使用编程语言描述协议时，都将保留字段定义成字符串类型。然而，在后来的使用过程中，可能需要扩展的数据类型是结构体，或者扩展的数据项增多，原来定义的保留字段的个数不足时，造成一个保留字

段中存放多个数据或多个数据结构的组合。为消息双方实现消息处理增加了大量的代码，用于对保留字段中多个数据成员的解析，同时，还可能修改消息协议的描述。

如果使用 ASN.1 描述这些消息协议，将使用 ASN.1 中定义的数据类型代替编程语言中的数据类型来描述消息协议中的结构。可以将保留字段定义成一个有序成员待定结构类型的数据，称为扩展项集。扩展项集的成员是扩展项，扩展项是有序成员固定结构类型，扩展项可以简单包含两个成员：扩展项标识（对象标识类型）和扩展项信息（任意类型，是具体扩展数据的 DER 编码）。在消息协议使用过程中没有扩展项时，该扩展项集无须赋值。需要加入扩展项时，给扩展项集中一个扩展项赋值，即填充扩展项类型标识（表示是何种扩展项）和扩展项信息。有多个扩展项时，由于扩展项集是有序成员待定结构类型的数据，可以在使用时随意加入数据成员（加入多个扩展项）。这样无须增加对扩展项信息组合和解析的代码，双方也不必协商扩展项在保留字段中如何组合，也不受扩展项数据类型和个数的限制。同时，根据扩展项标识可以轻松获得接受方所需的扩展项。

ASN.1 抽象文法描述的优势，在应用系统消息协议中，特别是大型消息协议如电子商务体系中证书认证系统和支付系统协议等方面得到了发挥。使得 ASN.1 文法描述越来越得到系统设计和软件开发人员的一致认可，并且被更为广泛地应用。

4.2 BER（基本编码规则）与 DER（定长编码规则）

4.2.1 数据类型标识

4.2.1.1 简介

ASN.1 应用越来越广泛的主要原因之一是这种抽象文法描述与几种标准化编码规则联系在一起。这些编码规则规定了如何在非传输介质下实现 ASN.1 定义的数据类型与字节流的相互转换。因为编码规则是针对 ASN.1 描述的数据类型而制定的，因此可以称这些编码规则为 ASN.1 的编码规则。

ASN.1 的编码规则是把使用 ASN.1 语言说明的数据转化成一种标准格式的系列规则，同时，保证转换后的数据在任意操作系统中，只要使用相同编码规则的解码器就可以解码获得原始数据。可见，ASN.1 和相应的编码规则保证了数据在分布式计算机系统中传输的一致性。对同一个 ASN.1 描述的数据可以采用不同的编码规则，选择使用哪一种编码规则取决于协议设计者的设计初衷。

目前，ASN.1 的标准化的编码规则有以下几种：BER（Basic Encoding Rules），DER（Distinguished Encoding Rules），PER（Packed Encoding Rules）和 CER（Canonical Eneoding Rules）等。其中，BER 是 20 世纪 80 年代初制定的，被广泛用于应用系统中，如用于互联网管理的简单网络管理协议（Simple Network Management Protocal，SNMP），用于互传电子邮件的消息处理服务（Message Handling Service，MHS），以及用于控制计算机和电话交互信息使用的 TSAPI 等。DER 是 BER 编码的一种特殊形式，它专门适用于具有安全特性的应用系统，如涉及加密技术和要求编解码信息唯一性的系统，如电子商务系统。CER 与 DER 类似，是 BER 编码的另外一种特殊形式，CER 的最大特点是对大数据实现编码，而且在数据还未完全获得之前就可以进行编码。但是由于业界认定在安全传输中最好的编码

方法是 DER 编码，所以 CER 编码未得到广泛使用。PER 是最近制定的系列编码规则，它的显著特点是使用有效的算法对数据编码，获得比 DER 更快更紧凑的数据，因此 PER 常常被应用于带宽或 CPU 饿死状态的应用系统，如空中交通管制和视听通信等领域。

4.2.1.2　数据类型标识和派生数据类型标识

1．数据类型标识

ASN.1 的数据类型有原子类型和结构类型两类，在使用过程中根据需要还可以从原有的数据类型派生出新的类型，称为派生类型。在 ASN.1 定义的数据类型中除了 CHOICE 和 ANY 两种类型以外，其他每种类型都有一个唯一的类型标识（Tag），用于标识一种数据类型。由于 CHOICE 是在几种数据类型中任意选择一个数据类型的数据，因此 CHOICE 类型标识可以是被选择数据的数据类型的类型标识；ANY 是某一类型数据编码后的信息，因此无法定义其标识。

ASN.1 中数据类型的标识分为四类：

（1）通用类（Universal 类），此类标识的值在所有应用中的定义相同；

（2）应用类（Application 类），此类标识的值只为某一种应用定义；

（3）私有类（Private 类），此类标识的值是为某些企业或公司定义；

（4）上下文说明类（Context-Specific 类），此类标识的值是为某个特定的类型定义的；由于上下文中使用了相同的数据类型的数据，并且其中有可选项，为了区别上下文中相同的数据类型的具体赋值，将其中的一个数据类型通过派生的方式改变类型标识，改变后的类型标识的类型就是上下文说明类。

由于类型标识是对数据类型的唯一标识，所谓的唯一标识应该是通用类的类型标识。表 4-1 给出了常用数据类型的通用类标识。

表 4-1　常用数据类型的通用类标识

类型	标识（十进制）	标识（十六进制）
BOOLEAN	1	01
INTEGER	2	02
BIT STRING	3	03
OCTET STRING	4	04
NULL	5	05
OBJECT IDENTIFIER	6	06
UTF8String	12	0C
SEQUENCE 和 SEQUENCE OF	16	10
SET 和 SET OF	17	11
NumericString	18	12
PrintableString	19	13
T61String	20	14
IA5String	22	16
UTCTime	23	17
GeneralizedTime	24	18
BMPString	30	1E

2. 派生数据类型标识

在类型标识的上下文说明类中，如果某个结构类型的成员中有若干相邻的可选项成员，其中有两项成员数据相同，由于解码时无法区别是哪一个成员的类型标识，因此需要采用派生的方法改变其中一个数据成员的类型标识。派生的数据类型标识的类型是除了通用类以外的三种标识类型，一般情况是上下文说明类。派生数据类型标识的方法有两种：显式派生法和隐式派生法。

显式派生法在 ASN.1 描述中使用[[Class] Number] Explicit 作为关键字，用它声明的类型表示使用了显式派生法，派生后生成的 Class 类型的数据类型标识，其值为 Number 的数值。Class 的类型可以是通用类、应用类和私有类，但未做声明时表示是上下文说明类，通常情况下 Class 空缺。显式派生法派生的数据类型在编码过程中与原始类型数据的编码不同。对于显式派生数据类型的编码，首先使用原数据的通用类标识对数据进行编码，再使用派生数据类型的标识对原数据获得编码信息并进行二次编码。

隐式派生法在 ASN.1 描述中使用[[Class] Number] Implicit 作为关键字，用它声明的类型表示使用了隐式派生法，派生后生成的 Class 类型的数据类型标识，其值为 Number 的数值。Class 的类型可以是通用类、应用类和私有类，但未做声明时表示是上下文说明类，通常情况下 Class 空缺。隐式派生法派生的数据类型在编码过程中与原始类型数据的编码不同。隐式派生数据类型的编码是使用派生获得的数据类型标识代替原数据的通用类标识对数据进行编码，但编码规则还是原数据类型的编码规则。

显式派生法和隐式派生法与原数据编码的比较如图 4-1 所示。

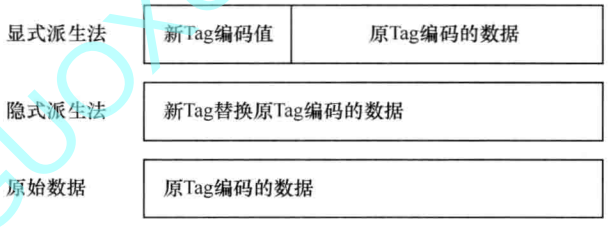

图 4-1　显式派生法与隐式派生法的对比

显式派生法用于任何数据类型的派生，特别适合于 CHOICE 和 ANY 类型，因为这两种类型没有自己的类型表示，使用隐式派生法无法替换原有数据类型标识。隐式派生法用于除了 ANY 类型和 CHOICE 类型以外的所有类型的派生。

4.2.2　BER 基本编码规则

BER 编码信息由以下几部分组成：

① 标识串，表示要编码的 ASN.1 类型的标识类和标识码以及使用的编码方法（是简单型还是结构型）。

② 长度串，定长型编码方法中它表示内容串的长度，非定长编码方法中它表示长度不定。

③ 内容串，简单定长型编码方法中它表示要编码类型值的具体内容，结构型编码方法中表示各个成员编码的串联。

④ 内容结束串，只有在结构非定长型编码方法中表示内容串的结束，其他方法中该串省略。