

Reemplazada por una versión más reciente



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Enmienda 1
X.680
(04/95)

**REDES DE DATOS Y COMUNICACIÓN
ENTRE SISTEMAS ABIERTOS**

**GESTIÓN DE REDES DE INTERCONEXIÓN DE
SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS
– NOTACIÓN DE SINTAXIS ABSTRACTA UNO**

**TECNOLOGÍA DE LA INFORMACIÓN –
NOTACIÓN DE SINTAXIS ABSTRACTA
UNO – ESPECIFICACIÓN DE LA
NOTACIÓN BÁSICA**

ENMIENDA 1: REGLAS DE EXTENSIBILIDAD

Enmienda 1 a la
Recomendación UIT-T X.680
Reemplazada por una versión más reciente

(Anteriormente «Recomendación del CCITT»)

Reemplazada por una versión más reciente

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. En el UIT-T, que es la entidad que establece normas mundiales (Recomendaciones) sobre las telecomunicaciones, participan unos 179 países miembros, 84 empresas de explotación de telecomunicaciones, 145 organizaciones científicas e industriales y 38 organizaciones internacionales.

Las Recomendaciones las aprueban los Miembros del UIT-T de acuerdo con el procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1993). Adicionalmente, la Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, aprueba las Recomendaciones que para ello se le sometan y establece el programa de estudios para el periodo siguiente.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI. El texto de la Recomendación UIT-T X.680, Enmienda 1, se aprobó el 10 de abril de 1995. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 8824-1.

NOTA

En esta Recomendación, la expresión «Administración» se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

© UIT 1996

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

Reemplazada por una versión más reciente

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

(Febrero de 1994)

ORGANIZACIÓN DE LAS RECOMENDACIONES DE LA SERIE X

Dominio	Recomendaciones
REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1-X.19
Interfaces	X.20-X.49
Transmisión, señalización y conmutación	X.50-X.89
Aspectos de redes	X.90-X.149
Mantenimiento	X.150-X.179
Disposiciones administrativas	X.180-X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Modelo y notación	X.200-X.209
Definiciones de los servicios	X.210-X.219
Especificaciones de los protocolos en modo conexión	X.220-X.229
Especificaciones de los protocolos en modo sin conexión	X.230-X.239
Formularios para enunciados de conformidad de implementación de protocolo	X.240-X.259
Identificación de protocolos	X.260-X.269
Protocolos de seguridad	X.270-X.279
Objetos gestionados de capa	X.280-X.289
Pruebas de conformidad	X.290-X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300-X.349
Sistemas móviles de transmisión de datos	X.350-X.369
Gestión	X.370-X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400-X.499
DIRECTORIO	X.500-X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600-X.649
Denominación, direccionamiento y registro	X.650-X.679
Notación de sintaxis abstracta uno	X.680-X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	X.700-X.799
SEGURIDAD	X.800-X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Cometimiento, concurrencia y recuperación	X.850-X.859
Tratamiento de transacciones	X.860-X.879
Operaciones a distancia	X.880-X.899
TRATAMIENTO ABIERTO DISTRIBUIDO	X.900-X.999

Reemplazada por una versión más reciente

ÍNDICE

Página

1	Alcance.....	1
2	Referencias normativas	1
	2.1 Recomendaciones Normas Internacionales idénticas.....	1
3	Modificaciones a la introducción	1
4	Modificaciones a las definiciones	2
5	Modelo ASN.1 de extensión de tipos	2
6	Requisitos de extensibilidad en reglas de codificación	3
7	Modificaciones de la definición de módulo	3
8	Modificación de la producción para ValueSet	4
9	Modificación de la definición de tipos y valores	4
10	Modificaciones de ENUMERATED.....	4
11	Modificaciones de SEQUENCE	5
12	Modificaciones de SET	5
13	Modificaciones de CHOICE	6
14	Modificaciones de tipos constreñidos	6
15	Modificaciones del identificador de excepción.....	6
16	Modificaciones de la especificación de conjunto de elementos	7
	Anexo A – Anexo explicativo del modelo ASN.1 de extensión de tipo	8

Reemplazada por una versión más reciente

Resumen

Las reglas de extensibilidad de la ASN.1 describen cómo escribir un módulo ASN.1 de modo que se pueda pasar por fases a una nueva versión de una especificación ASN.1.

Reemplazada por una versión más reciente

Introducción

Esta Recomendación | Norma Internacional documenta las modificaciones a la Rec. UIT-T X.680 | ISO/CEI 8824-1, que se precisan para soportar las reglas de extensibilidad ASN.1.

Las reglas de extensibilidad de la ASN.1 describen cómo escribir un módulo ASN.1 de modo que se pueda pasar por fases a una nueva versión de la especificación ASN.1. La nueva versión puede diferir de versiones anteriores por la adición de nuevos componentes a un SET (conjunto), SEQUENCE (secuencia), o CHOICE (elección), la adición de nuevas enumeraciones a un tipo enumerado, y la relajación de las limitaciones en una especificación de subtipo. Un paso por fases a una nueva versión de la especificación ASN.1 permite a las entidades pares que comunican a través de la red tener simultáneamente conocimiento diferente del conjunto de valores permitidos en la sintaxis abstracta y, sin embargo, poder comunicar aunque cualquier entidad par no sepa que su entidad par tiene una comprensión diferente de cuál es el conjunto de valores en la sintaxis abstracta.

Por ejemplo, inicialmente las entidades pares A,B,C y D pueden tener una visión idéntica de los tipos de valores que se pueden intercambiar. Suponiendo que la especificación ASN.1 que describe dichos valores se definió originalmente teniendo en cuenta la extensibilidad, puede extenderse añadiendo nuevos componentes a una SEQUENCE (secuencia), por ejemplo, creando de esa forma una nueva versión de la especificación ASN.1. Las entidades pares A y B pueden adoptar inmediatamente la nueva versión, mientras que las entidades pares C y D continúan utilizando la versión antigua, aunque las cuatro entidades pares pueden seguir comunicando entre sí sin dificultad porque la secuencia se definió originalmente como extensible. Cuando los pares A y B intercambian un valor de la secuencia recién extendida, ambos verán valores para los componentes nuevos que se añadieron a la secuencia (siempre que se envíen). Cuando la entidad par A (o B) envía un mensaje que contiene los componentes nuevos añadidos a la entidad par C (o D), el mensaje aparecerá para C en el nivel abstracto (es decir, después de decodificar el valor) sin ninguno de los nuevos componentes añadidos; sólo se verán en la secuencia los componentes originales. Cuando la entidad par C (o D) envía un mensaje a la entidad par A (o B), el mensaje (por supuesto) no contendrá ninguno de los componentes nuevos, sin embargo, parecerá a A como si una entidad par que utiliza la misma versión de la especificación ASN.1 que A (es decir, B) hubiera enviado un mensaje al que le faltan todos los componentes nuevos. Las entidades pares C y D continúan intercambiando mensajes entre sí como siempre lo hicieron.

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

**TECNOLOGÍA DE LA INFORMACIÓN – NOTACIÓN DE SINTAXIS
ABSTRACTA UNO – ESPECIFICACIÓN DE LA NOTACIÓN BÁSICA**

**ENMIENDA 1:
(a la Rec. X.680 | ISO/CEI 8824-1)**

Reglas de extensibilidad

1 Alcance

Esta Recomendación | Norma Internacional documenta las modificaciones a la Rec. UIT-T X.680 | ISO/CEI 8824-1 necesarias para soportar las reglas de extensibilidad ASN.1.

2 Referencias normativas

Las siguientes Recomendaciones | Normas Internacionales contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas Internacionales son objeto de revisiones, por lo que se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y Normas citadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes.

2.1 Recomendaciones | Normas Internacionales idénticas

- Recomendación UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Tecnología de la información – Notación de sintaxis abstracta uno – Especificación de la notación básica.*
- Recomendación UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Tecnología de la información – Notación de sintaxis abstracta uno – Especificación de constricciones.*
- Recomendación UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *Tecnología de la información – Notación de sintaxis abstracta uno – Parametrización de las especificaciones de notación de sintaxis abstracta uno.*
- Recomendación UIT-T X.690 (1994) | ISO/CEI 8825-1:1995, *Tecnología de la información – Reglas de codificación: Reglas de codificación básica, reglas de codificación canónica y reglas de codificación distinguida.*
- Recomendación UIT-T X.691 (1995) | ISO/CEI 8825-2:1995, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno – Reglas de codificación paquetizada.*

3 Modificaciones a la introducción

{Añádase el texto siguiente a la introducción de la Rec. UIT-T X.680 | ISO/CEI 8824-1 inmediatamente antes del párrafo que empieza con «Las cláusulas 8 a 31»:}

Una especificación ASN.1 se realizará inicialmente con un conjunto de tipos ASN.1 totalmente definidos. En una etapa posterior, sin embargo, puede ser necesario modificar estos tipos (habitualmente mediante la adición de componentes suplementarios en una secuencia o tipo de conjunto). Si esto ha de ser posible de manera que las implementaciones que

utilizan las definiciones antiguas de tipos puedan interfuncionar con implementaciones que utilicen las nuevas definiciones de tipos de una manera determinada, las reglas de codificación deberán dar el soporte necesario. La notación ASN.1 soporta la inclusión de un **marcador de extensión** en algunos tipos. Este indica a las reglas de codificación la intención del diseñador de que este tipo sea uno de una serie de tipos relacionados (es decir, versiones del mismo tipo inicial), denominada **serie de extensión**, y que las reglas de codificación deberán permitir la transferencia de información entre implementaciones que utilizan tipos diferentes que estén relacionados por formar parte de la misma serie de extensión.

4 Modificaciones a las definiciones

{Añádanse las nuevas definiciones siguientes a la Rec. UIT-T X.680 | ISO/CEI 8824-1, manteniendo el orden alfabético de las definiciones en la Rec. UIT-T X.680 | ISO/CEI 8824-1. Téngase en cuenta que el carácter alfabético que aparece en los números de cláusula siguientes se cambiará por un carácter numérico adecuado cuando se añadan las definiciones al documento de base:}

3.8.a serie de extensión: Serie de tipos de notación de sintaxis abstracta uno que puede ordenarse de manera que cada tipo sucesivo en la serie esté formado por adición de texto al final de la notación de tipo inmediatamente anterior en la serie.

NOTA – Pueden extenderse tanto los tipos anidados como los no anidados.

3.8.b marcador de extensión: Bandera sintáctica (una elipsis) que se incluye en todos los tipos que forman parte de una serie de extensión.

3.8.c raíz de extensión: Tipo extensible que es el primer tipo de una serie de extensión. Incluye el marcador de extensión sin notación adicional que no sea comentarios y espacios en blanco entre el marcador de extensión y el correspondiente «}» o «)».

NOTA – Sólo una raíz de extensión puede ser el primer tipo de una serie de extensión.

3.8.d adición de extensión: Una de las notaciones añadidas en una serie de extensión. Para tipos conjunto y tipos secuencia, cada adición de extensión es la adición de un elemento único. Para tipos enumerados es la adición de una enumeración ulterior única. Para tipos elección es la adición de una elección ulterior única. Para una restricción es la adición de un elemento subtipo.

NOTA – Las adiciones de extensión están ordenadas tanto textualmente (siguiendo al marcador de extensión) como lógicamente (con rútilos crecientes o valores de enumeración).

3.8.e relacionados por extensión: Dos tipos que tienen la misma raíz de extensión y en los que uno se generó añadiendo cero o más adiciones de extensión a la otra.

3.8.f tipo extensible: Tipo con un marcador de extensión.

3.8.g restricción extensible: Una restricción de subtipo con un marcador de extensión.

5 Modelo ASN.1 de extensión de tipos

{Añádase este texto como nueva cláusula antes de la cláusula 6 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, utilizando el mismo encabezamiento que en esta cláusula:}

Cuando se decodifica un tipo de extensión, el decodificador puede detectar:

- a) la ausencia de adiciones de extensión esperadas en un tipo secuencia o conjunto; o
- b) la presencia de adiciones de extensión arbitrarias no esperadas sobre las definidas (si existen) en un tipo secuencia o conjunto, o de una alternativa desconocida en un tipo elección, o una enumeración desconocida en un tipo enumerado, o de una longitud o valor no esperados de un tipo cuya restricción es extensible.

En términos formales, una sintaxis abstracta definida por un tipo extensible «X» contiene no sólo el valor del tipo «X», sino también los valores de todos los tipos que están relacionados por extensión a «X». De esta manera, el proceso de decodificación nunca indica un error cuando se detecta alguna de las situaciones anteriores (a o b). Las medidas a tomar en cada situación deben especificarse por el diseñador de la capa de aplicación.

NOTA – A menudo estas medidas consistirán en ignorar la presencia de extensiones adicionales no esperadas y en utilizar un valor por defecto o un indicador «missing» («falta») para adiciones de extensión esperadas que estén ausentes.

Las adiciones de extensión no esperadas detectadas por un decodificador en un tipo extensible pueden incluirse más tarde en una codificación subsiguiente de dicho tipo (para su transmisión en retorno al emisor o a un tercero), siempre que se utilice la misma sintaxis de transferencia en la transmisión subsiguiente.

6 Requisitos de extensibilidad en reglas de codificación

{Añádase este texto como nueva cláusula a la Rec. UIT-T X.680 | ISO/CEI 8824-1 antes de la cláusula 6 y después de la cláusula añadida anterior, utilizando el mismo encabezamiento que la presente cláusula:}

6.1 Todas las reglas de codificación ASN.1 permitirán la codificación de valores de un tipo extensible «X» de manera que puedan ser decodificados utilizando un tipo extensible «Y» que esté relacionado por extensión con «X». Además, las reglas de codificación permitirán que los valores decodificados mediante «Y» puedan recodificarse (utilizando «Y») y decodificarse mediante un tercer tipo extensible «Z» que esté relacionado por extensión con «Y» (y por lo tanto también con «X»).

NOTA – Los tipos «X», «Y» y «Z» pueden aparecer en cualquier orden en la serie de extensión.

Si un valor de un tipo extensible «X» se codifica y luego se retransmite (directamente o a través de una aplicación de retransmisión que utiliza un tipo relacionado por extensión «Z») a otra aplicación que decodifica el valor utilizando el tipo extensible «Y» que está relacionado por extensión con «X», el decodificador que utiliza el tipo «Y» obtiene un valor abstracto compuesto por:

- a) un valor abstracto del tipo raíz de extensión;
- b) un valor abstracto de cada adición de extensión que está presente en «X» y en «Y»;
- c) una codificación delimitada para cada adición de extensión (si existe) que está en «X» pero no en «Y».

Las codificaciones en c) podrán ser incluidas en una codificación posterior de un valor «Y», si la aplicación así lo exige. Dicha codificación debe ser una codificación válida de un valor «X».

Ejemplo aclaratorio: Si el sistema A utiliza un tipo raíz extensible (tipo «X») que es un tipo secuencia o un tipo conjunto con una adición de extensión de un tipo entero opcional, mientras que el sistema B utiliza un tipo relacionado por extensión (tipo «Y») que tiene dos adiciones de extensión cada una de ellas de tipo entero opcional, la transmisión por B de un valor de «Y» que omita el valor entero de la primera adición de extensión y que incluya el de la segunda no debe ser confundido por A con la presencia de (sólo) la primera adición de extensión de «X» que conoce. Además, A tiene que poder recodificar el valor de «X» con un valor presente para el primer tipo entero, seguido del segundo valor entero recibido de B, si así lo requiere el protocolo de aplicación.

6.2 Todas las reglas de codificación ASN.1 especificarán la codificación y decodificación del valor de un tipo enumerado y de un tipo elección de manera que, si un valor transmitido se encuentra en el conjunto de adiciones de extensión común al codificador y al decodificador, se decodifique con éxito, en otro caso el decodificador deberá poder delimitar su codificación e identificarlo como un valor de una adición de extensión (desconocida).

6.3 Todas las reglas de codificación ASN.1 especificarán los tipos de codificación y de decodificación con constricciones extensibles de manera que, si un valor transmitido está en el conjunto de adiciones de extensión común al codificador y al decodificador, se decodifique con éxito, en otro caso el decodificador deberá poder delimitar su codificación e identificarlo como un valor de una adición de extensión (desconocida).

En todos los casos, la presencia de adiciones de extensión no afectará la aptitud para reconocer información posterior cuando un tipo con un marcador de extensión está anidado dentro de algún otro tipo.

NOTA – Todas las variantes de las reglas de codificación básica de ASN.1 y de las reglas de codificación empaquetada de ASN.1 cumplen todos estos requisitos.

7 Modificaciones de la definición de módulo

{Modifíquense las producciones en 10.1 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 de la manera siguiente:}

```
ModuleDefinition ::=
    ModuleIdentifier
    DEFINITIONS
    TagDefault
    ExtensionDefault
    "::<="
```

```
BEGIN
ModuleBody
END
```

```
ExtensionDefault ::=
EXTENSIBILITY IMPLIED | empty
```

{Todas las demás producciones en 10.1 se mantienen sin modificación.}

{Inclúyase una nueva subcláusula después de 10.3 en la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

10.3 bis La opción «EXTENSIBILITY IMPLIED» es equivalente a la inserción textual de un marcador de extensión (...) en la definición de cada tipo en el módulo para el cual eso está permitido. La ausencia de «EXTENSIBILITY IMPLIED» significa que sólo se proporciona extensibilidad para aquellos tipos dentro del módulo en los que está explícitamente presente un marcador de extensión.

NOTA – «EXTENSIBILITY IMPLIED» afecta sólo tipos. No produce efecto sobre conjuntos de objetos.

8 Modificación de la producción para ValueSet

{Sustitúyase la producción para ValueSet en 13.5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 por:}

```
ValueSet ::= "{" ElementSetSpecs "}"
```

9 Modificación de la definición de tipos y valores

{Inclúyase una (nueva) subcláusula 14.12 en la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

14.12 La presencia implícita o explícita de un marcador de extensión en la definición de un tipo no produce efecto sobre la notación de valor. Es decir, la notación de valor para un tipo con marcador de extensión es exactamente la misma que si no estuviera el marcador de extensión.

10 Modificaciones de ENUMERATED

{Modifíquese la producción en la cláusula 17 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 para que diga:}

```
EnumeratedType ::= ENUMERATED "{" Enumerations "}"
Enumerations ::= RootEnumeration |
                RootEnumeration "," "..." |
                RootEnumeration "," "..." "," AdditionalEnumeration
```

```
RootEnumeration ::= Enumeration
```

```
AdditionalEnumeration ::= Enumeration
```

{«Enumeration» se define como lo está ahora.}

{Modifíquese la Nota 2 de 17.1 para que se lea:}

2 No es necesario que los valores numéricos dentro de los «NamedNumber» en la «RootEnumeration» estén ordenados o sean contiguos, y no es necesario que los valores numéricos dentro de los «NamedNumber» en la «AdditionalEnumeration» sean contiguos.

{Añádase el texto siguiente después de 17.3:}

17.3 bis El valor de cada nueva «AdditionalEnumeration» será mayor que el de todas las «AdditionalEnumeration» definidas anteriormente en el tipo.

17.3 ter Cuando se utiliza un «NamedNumber» en la definición de una «AdditionalEnumeration», los valores asociados con él serán diferentes del valor de todos los «EnumerationItem» definidos anteriormente (en este tipo), independientemente de que los «EnumerationItem» anteriormente definidos aparecieron o no en la raíz de enumeración. Por ejemplo:

```
A ::= ENUMERATED {a, b, ..., c(0)}           -- inválido, porque 'a' y 'b' son iguales a 1
B ::= ENUMERATED {a, b, ..., c, d(2)}       -- inválido, porque 'c' y 'd' son iguales a 2
C ::= ENUMERATED {a, b(3), ..., c(1)}       -- válido, 'c' = 1
D ::= ENUMERATED {a, b, ..., c(2)}         -- válido, 'c' = 2
```

17.3 quater El valor asociado con la primera alternativa de «AdditionalEnumeration» para la producción de «identifier» (no da «NameNumber» será el valor más pequeño para el cual a) no está definido un «EnumerationItem» en la «RootEnumeration» y b) todos los anteriores «EnumerationItem» en la «AdditionalEnumeration» (si existen) son más pequeños. Por ejemplo, todo lo que sigue es válido:

A ::= ENUMERATED {a, b, ..., c} -- c = 2
 B ::= ENUMERATED {a, b, c(0), ..., d} -- d = 3
 C ::= ENUMERATED {a, b, ..., c(3), d} -- d = 4
 D ::= ENUMERATED {a, z(25), ..., d} -- d = 1

11 Modificaciones de SEQUENCE

{Modifíquese la producción en la cláusula 22 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 para que diga:}

```
SequenceType ::= SEQUENCE "{" "|" |
                SEQUENCE "{" ExtensionAndException "}" |
                SEQUENCE "{" ComponentTypeLists "}"

ExtensionAndException ::= "... " | "... " ExceptionSpec

ComponentTypeLists ::= RootComponentTypeList |
                      RootComponentTypeList ", " ExtensionAndException |
                      RootComponentTypeList ", " ExtensionAndException ", " AdditionalComponentTypeList |
                      ExtensionAndException ", " AdditionalComponentTypeList

RootComponentTypeList ::= ComponentTypeList

AdditionalComponentTypeList ::= ComponentTypeList
```

{«ComponentTypeLists» se define como lo está ahora.}

NOTA – Los «ComponentType» que no están marcados OPTIONAL o DEFAULT en la «AdditionalComponentTypeList» deben siempre codificarse, excepto cuando el valor de datos de presentación se está retransmitiendo por un emisor que esté utilizando una versión anterior de la sintaxis abstracta en la que no está definido el «ComponentType».

{Sustitúyase la **primera oración** de 22.4 por la siguiente:}

«Type» en la notación «COMPONENTS OF Type» será un tipo secuencia que no contiene en sí mismo un marcador de extensión, pero puede tener componentes que sí lo contengan.

{Añádase la siguiente nueva cláusula después de 22.4:}

22.4 bis La notación «COMPONENTS OF Type» no se utilizará en «AdditionalComponentTypeList».

12 Modificaciones de SET

{Modifíquese la producción en la cláusula 24 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 para que diga:}

```
SetType ::= SET "{" "|" |
           SET "{" ExtensionAndException "}" |
           SET "{" ComponentTypeLists "}"
```

«ComponentTypeLists» se especifica en 22.1.

{Sustitúyase la primera frase de 24.2 por la siguiente:}

Si un «Type» contiene un marcador de extensión, ese «Type» se utiliza como el «Type» en un «COMPONENTS OF Type», el marcador de extensión no es visible para el «COMPONENTS OF Type». Así pues, para los fines del «COMPONENTS OF Type», el «Type» se trata como si no estuviese definido con un marcador de extensión.

{Añádase lo siguiente al texto después de 24.3:}

24.3 bis El rótulo de cada nuevo «ComponentType» añadido a la «AdditionalComponentTypeList» será canónicamente mayor (véase 6.4) que los de los otros componentes de la «AdditionalComponentTypeList», y aparecerá último en el tipo conjunto.

{Modifíquese la cláusula 24.4 para que diga:}

24.4 Las subcláusulas 22.2, 22.2 *bis* y 22.6 – 22.10 se aplican también a los tipos conjunto.

13 Modificaciones de CHOICE

{Modifíquese la producción en la cláusula 26 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 para que diga:}

ChoiceType ::= CHOICE "{" AlternativeTypeLists "}"

AlternativeTypeLists ::=

RootAlternativeTypeList |

RootAlternativeTypeList "," ExtensionAndException |

RootAlternativeTypeList "," ExtensionAndException "," AdditionalAlternativeTypeList

RootAlternativeTypeList ::= AlternativeTypeList

AdditionalAlternativeTypeList ::= AlternativeTypeList

{«AlternativeTypeList» se define como lo está ahora.}

{Añádase el texto siguiente después de 26.3:}

26.3 bis El rótulo de cada nuevo «NamedType» añadido a la «AdditionalAlternativeTypeList» será canónicamente mayor (véase 6.4) que los de los otros componentes de la «AdditionalAlternativeTypeList» y será el último «NamedType» en el tipo elección.

{Modifíquese 26.5, cambiando «Cuando este tipo se emplea ...» por:}

Cuando este tipo no tiene un marcador de extensión y se emplea ...

{Modifíquese 26.5 añadiendo lo siguiente al final de la primera oración:}

Cuando este tipo tiene un marcador de extensión, no se utilizará donde esta Norma Internacional requiera rótulos distintos (véanse 22.5, 24.3, 26.2).

14 Modificaciones de tipos constreñidos

{Sustituyanse las dos últimas oraciones de 42.5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, por:}

«ExceptionSpec» se define en la cláusula 43. A menos que se utilice junto con un «marcador de extensión» (véase la cláusula 44 *bis*), sólo estará presente si la «ConstraintSpec» incluye una ocurrencia de «DummyReference» (véase 8.4 de la Rec. UIT-T X.683 | ISO/CEI 8824-4) o es una «UserDefinedConstraint» (véase la Rec. UIT-T X.682 | ISO/CEI 8824-3, cláusula 9).

{Modifíquese la producción en 42.6 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 para que diga:}

SubtypeConstraint ::= ElementSetSpecs

15 Modificaciones del identificador de excepción

{Añádase una nueva subcláusula 43.5 a la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

43.5 Donde un tipo esté constreñido por constricciones múltiples, de las cuales más de una tengan un identificador de excepción, el identificador de excepción de la constricción más exterior será considerado como el identificador de excepción para ese tipo.

43.6 Donde esté presente un marcador de excepción sobre tipos que se utilizan en aritmética de conjuntos, el identificador de excepción en la constricción más externa se ignora y no es heredado por el tipo que se está constreñiendo como resultado de la aritmética de conjuntos.

16 Modificaciones de la especificación de conjunto de elementos

{Insértese inmediatamente antes de la producción «ElementSetSpec» en 44.1 de la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

```
ElementSetSpec ::=
  RootElementSetSpec |
  RootElementSetSpec "," "..." |
  "..." "," AdditionalElementSetSpec |
  RootElementSetSpec "," "..." "," AdditionalElementSetSpec
```

```
RootElementSetSpec ::= ElementSetSpec
```

```
AdditionalElementSetSpec ::= ElementSetSpec
```

NOTA – Los elementos que se referencian por «ElementSetSpecs» son la unión de los elementos referenciados por «RootElementSetSpec» y «AdditionalElementSetSpec».

{Añádase una nueva cláusula 44 bis y lo siguiente a la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

44 bis Marcador de extensión

NOTA – Al igual que la notación constricción en general, el marcador de extensión no produce efecto sobre algunas reglas de codificación ASN.1, como las reglas de codificación básica, pero sí lo produce en otras como las reglas de codificación empaquetada.

44.1 El marcador de extensión, elipsis, es una indicación de que se esperan adiciones de extensión. No establece enunciados de cómo deben manejarse dichas adiciones salvo que no deben tratarse como un error durante el proceso de decodificación.

44.2 La utilización conjunta de un marcador de extensión y de un identificador de excepción es una indicación de que se esperan adiciones de extensión y que por lo tanto no deben tratarse como un error en el proceso de decodificación y que las normas de aplicación prescriben que la aplicación tome medidas específicas si existe una violación de constricción. Se recomienda que esta notación se utilice en aquellas situaciones en las que se está utilizando el procedimiento de almacenamiento y retransmisión o cualquier otra forma de retransmisión, de manera que se indique que cualquier adición de extensión no reconocida se devuelva a la aplicación para una posible recodificación y retransmisión.

44.3 Si existe aritmética de conjuntos en la notación «ElementSetSpecs», se realizará sin tener en cuenta la presencia del marcador de extensión.

NOTA – En otras palabras, la presencia de un marcador de extensión no produce efecto en la aritmética de conjuntos.

44.4 Si un tipo definido con una constricción extensible se referencia en un «ContainedSubtype», el tipo definido después no hereda el marcador de extensión. Si se desea que el tipo definido después sea extensible, deberá añadirse explícitamente un marcador de extensión a su «ElementSetSpecs». Por ejemplo:

```
A ::= INTEGER (0..10, ...)      -- A es extensible
B ::= INTEGER (A)              -- B es inextensible
C ::= INTEGER (A, ...)         -- C es extensible
```

44.5 Si un tipo definido con una constricción extensible sufre una ulterior constricción con un «ElementSetSpecs» que no contiene un marcador de extensión, el tipo resultante es uno cuya constricción no es extensible. Por ejemplo:

```
A ::= INTEGER (0..10, ...)      -- A es extensible
B ::= A (2..5)                 -- B es inextensible
C ::= A                        -- C es extensible
```

44.6 Los componentes de un tipo conjunto, secuencia o elección que hayan sido constreñidos a estar ausentes no podrán estar presentes, independientemente de que el tipo conjunto, secuencia o elección sea un tipo extensible.

NOTA – Las constricciones de tipo interior no producen efecto sobre la extensibilidad.

Por ejemplo:

```
A ::= SEQUENCE {
  a  INTEGER
  b  BOOLEAN OPTIONAL,
  ...
}
B ::= A (WITH COMPONENTS {b ABSENT})  -- B is extensible, but 'b' shall not be
                                       -- present in any of its values
```

{Añádase el siguiente anexo didáctico a la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

Anexo A

Anexo explicativo del modelo ASN.1 de extensión de tipo

(Este anexo no parte integrante de la presente Recomendación | Norma Internacional)

A.1 Visión general

A.1.1 Puede ocurrir que un tipo ASN.1 evolucione en el tiempo desde un tipo **raíz de extensión** mediante una serie de extensiones denominadas **adiciones de extensión**.

A.1.2 Un tipo ASN.1 disponible para una implementación determinada puede ser el tipo raíz de extensión o el tipo raíz de extensión junto con uno más adiciones de extensión. Cada uno de estos tipos ASN.1 que contiene una adición de extensión también incluye todas las adiciones de extensión definidas con anterioridad.

A.1.3 Las definiciones de tipos ASN.1 de esta serie se dice que están **relacionadas por extensión** (véase la cláusula 3.8 para una definición más precisa de «relacionada por extensión») y se precisan reglas de codificación para codificar tipos relacionados por extensión de manera que si dos sistemas están utilizando dos tipos diferentes que están relacionados por extensión, las transmisiones entre los dos sistemas transfieran con éxito el contenido de la información de las partes de los tipos relacionados por extensión que son comunes a ambos sistemas. También se requiere que aquellas partes que no son comunes a ambos sistemas puedan ser delimitadas y retransmitidas (quizá a un tercero) en una transmisión posterior, siempre que se utilice la misma sintaxis de transferencia.

NOTA – El emisor puede estar utilizando un tipo que se encuentre antes o después en la serie de adiciones de extensión.

A.1.4 La serie de tipos obtenidos por adición progresiva a un tipo raíz se denomina **serie de extensión**. Para que las reglas de codificación tengan debidamente en cuenta las transmisiones de tipos relacionados por extensión (lo que puede necesitar más bits en la línea), dichos tipos (incluido el tipo raíz de extensión) necesitan estar marcados sintácticamente. La marca es una elipsis (...) que se denomina **marcador de extensión**.

EJEMPLO:

Tipo raíz	1ª extensión	2ª extensión
SEQUENCE { a INTEGER, ... }	SEQUENCE { a INTEGER, ..., b BOOLEAN, c INTEGER }	SEQUENCE { a INTEGER, ..., b BOOLEAN, c INTEGER, d REAL }

A.1.5 La especificación de un tipo raíz de extensión aparece antes/encima del marcador de extensión ASN.1 y las adiciones de extensión aparecen después/debajo de él.

A.1.6 Un tipo que tiene un marcador de extensión puede anidarse dentro de un tipo que no tenga ninguno, o puede anidarse dentro de un tipo en una raíz de extensión, o dentro de un tipo que es una adición de extensión. En estos casos las series de extensión se tratan independientemente y el tipo anidado con el marcador de extensión no produce efecto sobre el tipo dentro del cual está anidado. Sólo puede aparecer una elipsis (es decir, un marcador de extensión) en una construcción específica.

A.1.7 Un nuevo tipo en la serie de extensión se define en términos de una única adición de extensión al tipo anterior en la serie de extensión. Esto no impide que se hagan múltiples adiciones de extensión en publicaciones sucesivas de algunas Recomendaciones | Normas Internacionales.

A.1.8 Aunque lo normal será añadir las adiciones de extensión en el tiempo, el modelo ASN.1 y la especificación subyacentes no consideran el tiempo. Dos tipos están relacionados por extensión si uno puede ser acrecentado (o «crecido») a partir del otro mediante adiciones de extensión. Es decir, uno contiene todos los componentes del otro. Pueden existir tipos que tienen que ser acrecentados («crecidos») en la dirección opuesta (aunque esto es poco probable). Puede incluso ocurrir que, con el tiempo, un tipo *empieza* con muchas adiciones de extensión que se quitaron progresivamente! De lo único que se ocupan la ASN.1 y sus reglas de codificación es de si un par de especificaciones de tipo están relacionadas por extensión o no. Si lo están, entonces **todas** las reglas de codificación ASN.1 asegurarán el interfuncionamiento entre sus usuarios.

A.1.9 Empezamos con un tipo y luego decidimos si vamos a querer interfuncionar con versiones anteriores y tenemos que ampliarlo posteriormente. Si es así, incluimos el marcador de extensión **ahora**. Podemos posteriormente añadir adiciones de extensión al tipo sin modificar los bits en la línea para valores anteriores, y manejando de una manera definida los valores extendidos por sistemas anteriores. Es sin embargo importante destacar que la adición de un marcador de extensión a un tipo que no lo tenía (o la supresión de un marcador de extensión) **cambiará** generalmente los bits en la línea e impedirá el interfuncionamiento. Dicho cambio requiere normalmente un cambio en el número de versión en todos los protocolos afectados.

A.1.10 El cuadro siguiente muestra los tipos ASN.1 que pueden formar el tipo raíz de extensión de una serie de extensión ASN.1 y la naturaleza de la adición de extensión única permitida para este tipo (por supuesto se pueden hacer múltiples adiciones de extensión en sucesión o juntas).

Adiciones de extensión

Tipo raíz de extensión	Naturaleza de la adición de extensión
ENUMERATED	Adición de una enumeración ulterior única al final de las «AdditionalEnumeration» con un valor de enumeración mayor que el de cualquier enumeración ya añadida.
SEQUENCE y SET	Adición de un «Component» único al final de la «AdditionalComponentTypeList». Obsérvese que no es obligatorio marcar el «ComponentType» añadido con OPTIONAL o DEFAULT, aunque normalmente se hace.
CHOICE	Adición de un «NamedType» único al final de la «AdditionalAlternativeTypeList».
Notaciones restricción	Adición de una única «AdditionalElementSetSpec» a la notación «ElementSetSpec».

A.2 Efectos sobre el número de la versión, etc.

A.2.1 Cuando se hace una nueva publicación de una especificación ASN.1 en la que las definiciones de tipo han cambiado con respecto a las definiciones de tipos relacionados con la extensión, entonces, para todos efectos, estos cambios, por sí mismos, no requieren un cambio en el identificador de objeto del módulo del número de versión del protocolo.

A.2.2 Puede darse el caso de que, por otros motivos, tales cambios pudieran ir acompañados de cambios en el número de la versión, pero esto no es obligatorio.

A.2.3 Por el contrario, la adición de un marcador de extensión a un tipo que antes no lo tenía, o la adición de componentes a un tipo secuencia o conjunto (por ejemplo) sin ningún marcador de extensión, crea un nuevo tipo que **no** está relacionado con el tipo antiguo en lo que respecta a la extensión; en este caso se debe dar un nuevo identificador de objeto al módulo que lo contiene, y sería conveniente dar un nuevo número de versión al protocolo asociado.

A.3 Requisitos sobre reglas de codificación

A.3.1 Se puede definir una sintaxis abstracta como los valores de un tipo ASN.1 único que sea un tipo extensible. Incluye entonces todos los valores que pueden obtenerse por adición o supresión de adiciones de extensión. Esta sintaxis abstracta se denomina una sintaxis abstracta relacionada por extensión.

A.3.2 Un conjunto de reglas de codificación bien formado para una sintaxis abstracta relacionada por extensión cumple los requisitos adicionales siguientes:

NOTA – Todas las reglas de codificación ASN.1 cumplen estos requisitos.

A.3.3 La definición de los procedimientos para transformar un valor abstracto en una codificación para transferencia, y para transformar una codificación recibida en un valor abstracto reconocerá la posibilidad de que el emisor y el receptor estén utilizando sintaxis abstractas que no son idénticas pero que están relacionadas por extensión.

A.3.4 En este caso, las reglas de codificación asegurarán que cuando el emisor tenga una especificación de tipo que sea anterior, en la serie de extensión, a la del receptor, los valores del emisor se transferirán al receptor de manera que éste pueda determinar que no hay adiciones de extensión.

A.3.5 Las reglas de codificación asegurarán que, cuando el emisor tenga una especificación de tipo que sea anterior, en la serie de extensión, a la del receptor, será posible la transferencia de valores de este tipo al receptor.

