



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.680

Enmienda 2
(06/99)

SERIE X: 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 2: Modelo semántico de notación de
sintaxis abstracta uno**

Recomendación UIT-T X.680 – Enmienda 2

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

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 declaraciones 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 de transmisión de datos por satélite	X.350–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.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–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

Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
SEGURIDAD	X.800–X.849

APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS

Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899

PROCESAMIENTO DISTRIBUIDO ABIERTO

X.900–X.999

Para más información, véase la Lista de Recomendaciones del UIT-T.

NORMA INTERNACIONAL 8824-1

RECOMENDACIÓN UIT-T X.680

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

ENMIENDA 2

Modelo semántico de notación de sintaxis abstracta uno

Resumen

Se espera que la enmienda 2 a la Rec. UIT-T X.680 | ISO/CEI 8824-1 sea de utilidad sobre todo para los constructores de herramientas. Tiene por objeto asegurar que todos ellos interpretan la norma ASN.1 de forma idéntica en lo que concierne a la correspondencia de valores y tipos. Especifica:

- a) qué notaciones ASN.1 son legales y cuáles no lo son;
- b) el valor exacto que cada nombre de referencia de valor identifica; y
- c) el conjunto exacto de valores que cada tipo o nombre de referencia de conjunto de valores identifica.

Orígenes

La enmienda 2 a la Recomendación UIT-T X.680 se aprobó el 18 de junio de 1999. Su texto se publica también, en forma idéntica, como ISO/CEI 8824-1.

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. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT.

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.

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.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2000

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.

ÍNDICE

	<i>Página</i>
1) Subcláusula 3.8.....	1
2) Nueva subcláusula 5.9	1
5.9 Referencias de valor y tipificación de valores.....	1
3) Nueva subcláusula 13.6	2
4) Nueva subcláusula 13.7	2
5) Subcláusula 15.2.....	2
6) Subcláusula 36.7.....	2
7) Subcláusula 48.3.2.....	2
8) Subcláusula 48.5.2.....	2
9) Subcláusula 48.8.2.....	2
10) Nuevo anexo F.....	3
Anexo F – Reglas para la compatibilidad de tipos y de valores	3
F.1 Necesidad del concepto de correspondencia de valores (introducción didáctica).....	3
F.2 Correspondencias de valores	5
F.3 Definiciones de tipo idénticas	6
F.4 Especificación de correspondencias de valores.....	7
F.5 Correspondencias de valores adicionales definidas para los tipos de cadenas de caracteres	8
F.6 Requisitos específicos de compatibilidad de tipo y de valor.....	9
F.7 Ejemplos.....	9

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

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

ENMIENDA 2

Modelo semántico de notación de sintaxis abstracta uno

1) Subcláusula 3.8

*Añádase las definiciones 3.8.39 bis y 3.8.71 bis a la subcláusula 3.8, y sustitúyase la definición de **gobernante; gobernador** por el siguiente nuevo 3.8.39:*

3.8.39 gobernante; gobernador (tipo): Definición, o referencia a, tipo que afecta a la interpretación de una parte de la sintaxis ASN.1, lo que exige que haga referencia a valores del tipo gobernador.

3.8.39 bis definiciones de tipo idénticas: Dos ejemplares de producción "Type" ASN.1 (véase la cláusula 16) se consideran definiciones de tipo idénticas si, tras realizar las transformaciones especificadas en el anexo F, se convierten en listas ordenadas idénticas de ítems ASN.1 (véase la cláusula 11).

3.8.71 bis correspondencia de valores: Relación de uno a uno entre valores de dos tipos que permite utilizar la referencia a uno de esos valores como referencia al otro valor. Puede utilizarse, por ejemplo, para especificar subtipos y valores por defecto (véase el anexo F).

Añádase a la subcláusula 3.8.50:

, y que gobierna la notación de subtipo.

2) Nueva subcláusula 5.9

Añádase una nueva subcláusula 5.9 como sigue:

5.9 Referencias de valor y tipificación de valores

5.9.1 ASN.1 define una notación de asignación de valor que permite dar un nombre a un valor de un tipo especificado. Este nombre puede utilizarse cuando se necesite una referencia a dicho valor. En el anexo F se describe y especifica el mecanismo de **correspondencia de valores** que permite que un nombre de referencia de valor asignado a un tipo de valor determinado pueda identificar un valor de otro tipo. De esta forma puede utilizarse la referencia al primer valor cuando se requiera una referencia a un valor del segundo tipo.

5.9.2 En la parte principal de las normas ASN.1 se utiliza texto en inglés corriente para especificar la legalidad (o la no legalidad) de construcciones en las que intervienen más de un tipo, pero los dos tipos deben ser "compatibles". Por ejemplo, el tipo utilizado para definir una referencia de valor debe ser "compatible con" el tipo gobernador cuando se usa la referencia de valor. En el anexo F, normativo, se utiliza el concepto de correspondencia de valores para hacer declaraciones precisas sobre la legalidad o ilegalidad de una determinada construcción ASN.1.

3) Nueva subcláusula 13.6

Añádase una nueva subcláusula 13.6 como sigue:

13.6 Cuando se utiliza un "DefinedType" como parte de una notación gobernada por un "Type" (por ejemplo, en una "SubtypeElementSpec"), el "DefinedType" deberá ser compatible con el "Type" gobernante, tal como se especifica en F.6.2.

4) Nueva subcláusula 13.7

Añádase una nueva subcláusula 13.7 como sigue:

13.7 Siempre que aparece un "DefinedValue" en una especificación ASN.1, está gobernado por un "Type", y ese "DefinedValue" deberá hacer referencia a un valor de un tipo que sea compatible con el "Type" gobernante, tal como se especifica en F.6.2.

5) Subcláusula 15.2

En 15.2, modifíquese la última frase introduciendo inmediatamente antes de "deberá" el siguiente texto:

está gobernado por "Type" y

6) Subcláusula 36.7

Añádase lo siguiente al ejemplo de la NOTA 3 de 36.7:

Una definición alternativa y no ambigua de "mystring" podría ser:

mystring MyAlphabet(BasicLatin) ::= "HOPE"

Formalmente, "mystring" es una referencia a un valor de un subconjunto de "MyAlphabet", pero según las normas de correspondencia de valores del anexo F puede utilizarse cuando se necesite una referencia a este valor dentro de "MyAlphabet".

7) Subcláusula 48.3.2

Sustitúyase 48.3.2 por el siguiente texto:

48.3.2 Un "ContainedSubtype" especifica todos los valores del tipo progenitor (parent) que están también en "Type". Es necesario que "Type" sea compatible con el tipo progenitor, tal como se especifica en F.6.3.

8) Subcláusula 48.5.2

En 48.5.2, suprimase la frase que dice " , o a tipos formados por rotulación a partir de cualquiera de estos tipos".

9) Subcláusula 48.8.2

En 48.8.2, suprimase la frase que dice " , o a los tipos formados por rotulación a partir de ellos".

10) Nuevo anexo F

Añádase un nuevo anexo F como sigue:

Anexo F

Reglas para la compatibilidad de tipos y de valores

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

El presente anexo se ha concebido principalmente para uso de los constructores de herramientas, y tiene por objeto garantizar que interpretan el lenguaje de forma idéntica. Se trata de especificar claramente qué ASN.1 son legales y cuáles no lo son, y de poder especificar el valor exacto que cada nombre de referencia de valor identifica, y el conjunto exacto de valores que cada nombre de referencia de conjunto de tipos o de valores identifica. El objetivo no es proporcionar la definición de transformaciones válidas de notaciones ASN.1 para otros propósitos que no sean los mencionados más arriba.

F.1 Necesidad del concepto de correspondencia de valores (introducción didáctica)

F.1.1 Considérense las siguientes definiciones ASN.1:

```
A ::= INTEGER
B ::= [1] INTEGER
C ::= [2] INTEGER (0..6,...)
D ::= [2] INTEGER (0..6,...,7)
E ::= INTEGER (7..20)
F ::= INTEGER {red(0), white(1), blue(2), green(3), purple(4)}
a A ::= 3
b B ::= 4
c C ::= 5
d D ::= 6
e E ::= 7
f F ::= green
```

F.1.2 Es evidente que las referencias de valor a, b, c, d, e y f pueden utilizarse en una notación de valor gobernada por A, B, C, D, E y F respectivamente. Por ejemplo:

```
W ::= SEQUENCE {w1 A DEFAULT a}
```

y:

```
x A ::= a
```

y:

```
Y ::= A(1..a)
```

son válidas teniendo en cuenta las definiciones de F.1.1. Sin embargo, si A fuera reemplazada por B, o C, o D, o E o F, ¿serían ilegales los enunciados resultantes? De igual forma, si la referencia de valor a fuera reemplazada en cada uno de estos casos por b, o c, o d, o e o f, ¿serían legales los enunciados resultantes?

F.1.3 Una cuestión más compleja sería considerar en cada caso la sustitución de la referencia de tipo por el texto explícito a la derecha de su asignación. Considérese por ejemplo:

```
f INTEGER {red(0), white(1), blue(2), green(3), purple(4)} ::= green
W ::= SEQUENCE {
    w1 INTEGER {red(0), white(1), blue(2), green(3), purple(4)}
    DEFAULT f}
x INTEGER {red(0), white(1), blue(2), green(3), purple(4)} ::= f
Y ::= INTEGER {red(0), white(1), blue(2), green(3), purple(4)}(1..f)
```

¿Serían estas notaciones ASN.1 legales?

F.1.4 Algunos de los ejemplos anteriores, aun siendo legales (como lo son la mayoría de ellos – véase el texto a continuación), son casos en los que no puede aconsejarse a los usuarios que escriban textos similares, porque son, como mínimo, oscuros, y en el peor de los casos confusos. Sin embargo, con frecuencia se usa una referencia a un valor de algún tipo [no necesariamente un tipo INTEGER (ENTERO)] como valor por defecto para ese tipo con la rotulación o la subtificación aplicada en el gobernador. El concepto de **correspondencia de valores** se ha introducido con el fin de proporcionar un medio claro y preciso de determinar qué construcciones – como las anteriores – son legales.

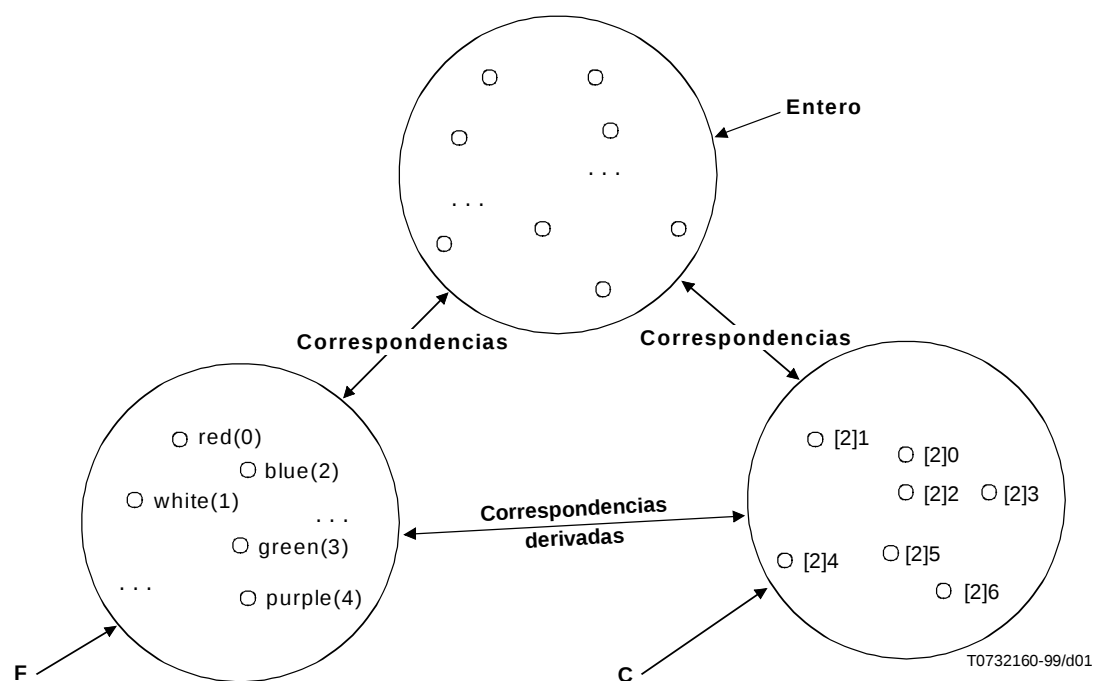
F.1.5 Considérese de nuevo:

```

C ::= [2] INTEGER (0..6,...)
E ::= INTEGER (7..20)
F ::= INTEGER {red(0), white(1), blue(2), green(3), purple(4)}

```

En cada caso se crea un nuevo tipo. Para F se puede identificar claramente una correspondencia de 1-1 entre sus valores y los valores del tipo universal "INTEGER". En el caso de C y E se puede identificar claramente una correspondencia de 1-1 entre sus valores y un subconjunto de los valores de tipo universal "INTEGER". Esta relación se denomina **correspondencia de valores** entre valores de los dos tipos. Además, puesto que entre los valores de F, C y E y los de "INTEGER" existe una correspondencia de (1-1), se puede utilizar esa relación para establecer la correspondencia entre los propios valores de F, C y E. Esto se ilustra en la figura F.1 para F y C.

**Figura F.1****F.1.6** Ahora, cuando se tiene una referencia de valor como:

```
c C ::= 5
```

para un valor de "C" requerido en algún contexto para identificar un valor de "F", siempre que exista una correspondencia de valores entre dicho valor de "C" y un (único) valor de "F" se puede definir "c" como referencia legal al valor de "F". Esto se ilustra en la figura F.2, en la que la referencia de valor "c" se utiliza para identificar un valor de "F", y puede utilizarse en lugar de una referencia directa f1 que de otro modo habría que definir:

```
f1 F ::= 5
```

F.1.7 Se señala que en algunos casos habrá valores de un tipo (7 a 20 de A en F.1.1, por ejemplo) que tengan correspondencias con otros valores de otro tipo (7 a 20 de E en F.1.1, por ejemplo), pero habrá otros valores que no las tengan (de 21 en adelante, de A). Las referencias a tales valores de A no proporcionarían una referencia válida a valores en E. (En este ejemplo, todo el conjunto E tiene una correspondencia de valores con un subconjunto de A. En general, puede haber un subconjunto de valores de ambos tipos que tengan correspondencias, mientras que otros valores de ambos tipos no la tengan.)

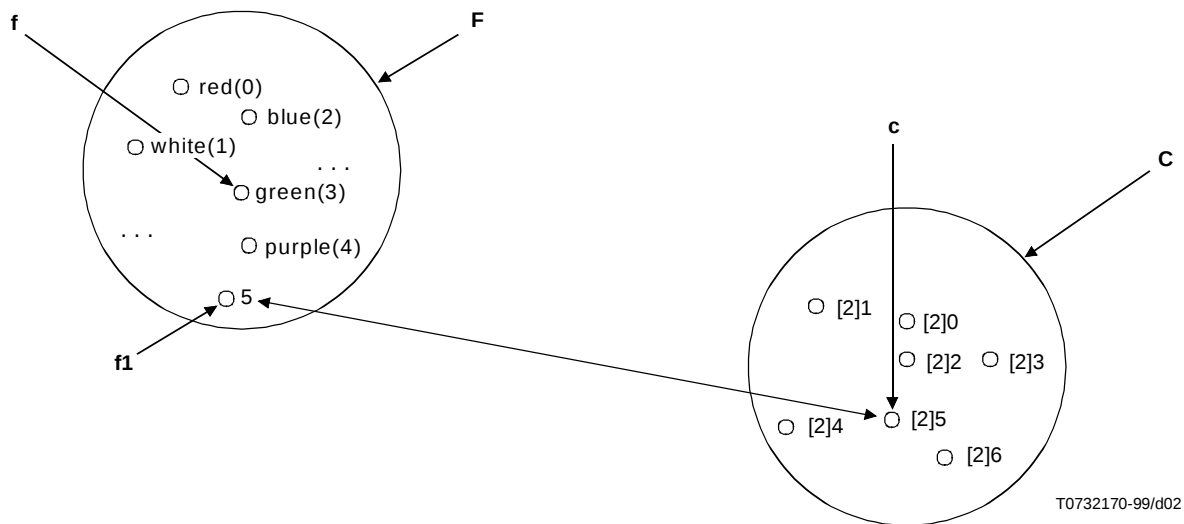


Figura F.2

F.1.8 En la parte principal de las normas ASN.1 se utiliza texto en inglés corriente para especificar la legalidad de los casos anteriores y casos similares. En la subcláusula F.6 se indican los requisitos exactos de legalidad, a los que habrá que remitirse siempre que existan dudas sobre una construcción compleja.

NOTA – El hecho de que se defina la existencia de correspondencias entre dos casos de la construcción "Type" permite utilizar referencias de valor establecidas usando una construcción "Type" para identificar valores en otra construcción "Type" que se asemeje lo suficiente a la anterior. Ello permite tipificar parámetros ficticios y reales utilizando dos construcciones "Type" textualmente separadas sin quebrantar las reglas de compatibilidad de los parámetros ficticios y reales. Y permite además especificar campos de clases de objeto de información utilizando una construcción "Type", y especificar el valor correspondiente en un objeto de información utilizando una construcción "Type" distinta que se asemeje lo suficiente. (Estos ejemplos no pretenden ser exhaustivos.) No obstante, se recomienda que se aproveche esta libertad únicamente en casos simples como "SEQUENCE OF INTEGER", o "CHOICE {int INTEGER, id OBJECT IDENTIFIER}", y no para construcciones "Type" más complejas.

F.2 Correspondencias de valores

F.2.1 El modelo fundamental consiste en tipos que, a modo de contenedores que no se superponen, contienen valores de tal forma que cada vez que aparece la construcción "Type" ASN.1 se define un nuevo tipo distinto (véanse las figuras F.1 y F.2). En el presente anexo se especifica cuándo existen **correspondencias de valores** entre esos tipos, permitiendo la utilización de una referencia a un valor de un tipo cuando se necesita una referencia a un valor en cualquier otro tipo.

Ejemplo: Considérese:

```
X ::= INTEGER
Y ::= INTEGER
```

X e Y son nombres de referencia de tipo (punteros) de dos tipos diferentes, pero existen correspondencias de valores entre estos dos tipos, por lo que puede utilizarse cualquier referencia a un valor de X cuando gobierna Y (por ejemplo, después de DEFAULT).

F.2.2 En el conjunto de todos los valores ASN.1 posibles, una correspondencia pone en relación dos valores. El conjunto completo de correspondencias de valores es una relación matemática. Dicha relación posee las siguientes propiedades: es reflexiva (cada valor ASN.1 está en relación consigo mismo), es simétrica (si se ha definido la existencia de una correspondencia entre un valor x1 y un valor x2, también existe, de manera automática, una correspondencia entre el valor x2 y el valor x1), y es transitiva (si existe una correspondencia entre el valor x1 y el valor x2, y otra correspondencia entre el valor x2 y el valor x3, existe automáticamente una correspondencia entre el valor x1 y el valor x3).

F.2.3 Además, dados dos tipos X1 y X2, considerados conjuntos de valores, el conjunto de correspondencias entre los valores de X1 y los valores de X2 es una relación de uno a uno, es decir, que para todos los valores x1 de X1 y x2 de X2, si existe una correspondencia entre x1 y x2:

- no existe ninguna correspondencia entre x1 y otro valor de X2 diferente de x2; y
- no existe ninguna correspondencia entre ningún valor de X1 (distinto de x1) a x2.

F.2.4 Cuando existe una correspondencia entre un valor x1 y un valor x2, la referencia de valor a cualquiera de los dos puede utilizarse automáticamente como referencia al otro si así lo requiere un tipo gobernante.

NOTA – El hecho de que se defina la existencia de correspondencias entre valores de algunas construcciones "Type" tiene únicamente por objeto proporcionar flexibilidad en la utilización de la notación ASN.1. **La existencia de tales correspondencias no implican en absoluto que los dos tipos tengan la misma semántica de aplicación**, pero se **recomienda** que las construcciones ASN.1 que pudieran ser ilegales sin correspondencias de valores se utilicen únicamente si los tipos correspondientes llevan realmente la misma semántica de aplicación. Se señala que con frecuencia existirán correspondencias de valores en cualquier especificación amplia entre dos tipos que sean construcciones ASN.1 idénticas, pero que lleven semánticas de aplicación completamente diferentes, y cuando la existencia de estas correspondencias de valores no se utilice nunca para determinar la legalidad de la especificación completa.

F.3 Definiciones de tipo idénticas

F.3.1 El concepto de **definiciones de tipo idénticas** se utiliza para hacer posible la definición de correspondencias de valores entre dos ejemplares de "Type" que sean idénticos o se asemejen lo suficiente como para que pueda esperarse razonablemente un uso intercambiable. Con el fin de precisar el significado de la expresión "se asemejen lo suficiente", en la presente subcláusula se especifica una serie de transformaciones que se aplican a cada uno de los ejemplares de "Type" para producir una **forma normal** de dichos ejemplares. Se dice que las definiciones de los dos ejemplares de "Type" son definiciones de tipo idénticas única y exclusivamente cuando sus formas normales son listas ordenadas idénticas de los mismos ítems ASN.1 (véase la cláusula 11).

F.3.2 Siempre que aparece "Type" en una especificación ASN.1, se trata de una lista ordenada de ítems definidos en la cláusula 11. La forma normal se obtiene aplicando las transformaciones definidas en F.3.2.1 a F.3.2.4, en ese orden.

F.3.2.1 Se eliminarán todos los comentarios (véase 11.6).

F.3.2.2 Las siguientes transformaciones no son recursivas y, por consiguiente, sólo es necesario aplicarlas una vez, en cualquier orden:

- Para cada tipo entero: "NamedNumberList" (véase 18.1), si existe, se reordena de forma que los "identifiers" estén en orden alfabético ("a" en primer lugar y "z" en último lugar).
- Para cada tipo enumerado: se añaden números, tal como se indica en 19.3, a cualquier "EnumerationItem" (véase 19.1) que sea un "identifier" (sin número); a continuación, "RootEnumeration" se reordena de forma que los "identifiers" estén en orden alfabético ("a" en primer lugar, y "z" en último lugar).
- Para cada tipo cadena de bits: "NamedBitList" (véase 21.1), si existe, se reordena de forma que los "identifiers" estén en orden alfabético ("a" en primer lugar, y "z" en último lugar).
- Para cada valor identificador de objeto: cada "ObjectIdComponent" se transforma en sus "NumberForm" correspondientes de conformidad con la semántica de la cláusula 31 (véase el ejemplo de 31.11).
- Para tipos secuencia (véase la cláusula 24) y tipos conjunto (véase la cláusula 26): cualquier extensión de la forma "ExtensionAndException", "ExtensionAdditions", se quita y coloca al final de "ComponentTypeLists"; se elimina "OptionalExtensionMarker", si está presente.

Si "TagDefault" es "IMPLICIT TAGS", se añade la palabra clave "IMPLICIT" a todos los ejemplares de "Tag" (véase la cláusula 30) a menos que:

- ya esté presente; o
- el ítem de palabra reservada EXPLICIT esté presente; o
- el tipo que se esté rotulando sea un tipo CHOICE; o
- se trate de un tipo abierto.

Si "TagDefault" es "AUTOMATIC TAGS", la decisión de si se aplica o no la rotulación automática se adopta de conformidad con 24.2 (la rotulación automática se realizará más adelante).

NOTA – En 24.3 y 26.2 se especifica que la presencia de un ítem "Tag" en un "ComponentType" que haya sido insertado como resultado de la sustitución de "Components of Type" no impide en sí misma la transformación por rotulación automática.

Si "ExtensionDefault" es "EXTENSIBILITY IMPLIED", se añade una elipsis ("...") después de "ComponentTypeLists", si no está presente.

- f) Para tipos elección (véase la cláusula 28): "RootAlternativeTypeList" se reordena de forma que los "identifiers" de "NameType" estén en orden alfabético ("a" en primer lugar, y "z" en último lugar). Si "TagDefault" es "IMPLICIT TAGS", se añade la palabra clave "IMPLICIT" a todos los ejemplares de "Tags" (véase la cláusula 30) a menos que:

- ya esté presente; o
- el ítem de palabra reservada EXPLICIT esté presente; o
- el tipo que se esté rotulando sea un tipo CHOICE; o
- se trate de un tipo abierto.

Si "TagDefault" es "AUTOMATIC TAGS", la decisión de si se aplica o no la rotulación automática se adopta de conformidad con 28.3 (la rotulación automática se realizará más adelante). Si "ExtensionDefault" es "EXTENSIBILITY IMPLIED", se añade una elipsis ("...") después de "AlternativeTypeLists", si no está presente.

F.3.2.3 Las siguientes transformaciones se aplicarán recursivamente en el orden indicado, hasta que se alcance un punto fijo:

- a) Para cada valor de identificador de objeto (véase 31.3): si la definición de valor empieza con un "DefinedValue", éste se sustituye por su definición.
- b) Para tipos secuencia y tipos conjunto: todos los ejemplares de "COMPONENTS OF Type" (véase la cláusula 24) se transforman de conformidad con las cláusulas 24 y 26.
- c) Para tipos secuencia, tipos conjunto y tipos elección: si anteriormente se ha decidido realizar la rotulación automática [véanse los ítems e) y f) de F.3.2.2], ésta se aplicará de conformidad con las cláusulas 24, 26 y 28.
- d) Para tipo elección: la construcción se sustituye por la alternativa seleccionada de conformidad con la cláusula 29.
- e) Todas las referencias de tipos son sustituidas por sus definiciones, de acuerdo a las siguientes reglas:
 - Si el tipo que se sustituye es una referencia al tipo que se está transformando, dicha referencia se sustituye por un ítem especial que se corresponde únicamente consigo mismo.
 - Si el tipo que se sustituye es un tipo secuencia de o un tipo conjunto de, las constricciones que siguen al tipo sustituido, si existen, se desplazan frente a la palabra clave "OF".
 - Si el tipo sustituido es un tipo parametrizado o un conjunto de valores parametrizados (véase 8.2 de la Rec. UIT-T X.683 | ISO/CEI 8824-4), cada ítem "DummyReference" es sustituido por el "ActualParameter" correspondiente.
- f) Todas las referencias de valores se sustituyen por sus definiciones; si el valor sustituido es un valor parametrizado (véase 8.2 de la Rec. UIT-T X.683 | ISO/CEI 8824-4), cada ítem "DummyReference" es sustituido por el "ActualParameter" correspondiente.

NOTA – Antes de sustituir cualquier referencia de valor, se aplicarán los procedimientos de este anexo para asegurar que la referencia de valor identifica, mediante correspondencias de valores o directamente, un valor en su tipo gobernante.

F.3.2.4 Para el tipo conjunto: "RootComponentTypeList" se reordena de forma que "ComponentType" estén en orden alfabético ("a" en primer lugar, y "z" en último lugar).

NOTA – El texto de 11.9 (bstring), 11.10 (hstring) y 11.11 (cstring) especifica que los caracteres de nuevo renglón y los espacios en blanco no son significativos en tales ítems. Si dos ocurrencias de dichos ítems contienen caracteres de nuevo renglón y espacio en blanco con diferentes usos, se tratan como ítems idénticos a los efectos de F.3.3.

F.3.3 Si dos ejemplares de "Type", al ser transformados en su forma normal, se convierten en listas idénticas de ítems ASN.1 (véase la cláusula 11), se dice que las definiciones de esos dos ejemplares de "Type" son definiciones de tipo idénticas con la siguiente excepción: si una "objectclassreference" (véase 7.1 de la Rec. UIT-T X.681 | ISO/CEI 8824-2) aparece dentro de la forma normalizada del "Type", **no** se considera que las definiciones de los dos tipos son definiciones de tipo idénticas, y no existirán correspondencias de valores (véase F.4) entre ellos.

NOTA – Se ha introducido esta excepción para evitar la necesidad de proporcionar reglas de transformación a forma normal en el caso de elementos de sintaxis que tengan que ver con notación de clase de objeto de información, objeto de información y conjunto de objetos de información. De igual forma, no se ha incluido en este momento una especificación de la normalización de toda la notación de valor y de la notación aritmética de conjunto. Si dicha especificación se requiriera, podría proporcionarse en una versión futura de la presente Recomendación | Norma Internacional. Se han introducido los conceptos de definiciones de tipo idénticas y de correspondencia de valores con el fin de que construcciones ASN.1 sencillas puedan utilizarse con nombres de referencia o copiando texto. Se ha considerado innecesario proporcionar esta funcionalidad para ejemplares más complejos de "Type" que incluyan clases de objeto de información, etc.

F.4 Especificación de correspondencias de valores

F.4.1 Si dos ocurrencias de "Type" son definiciones de tipo idénticas según las reglas de F.3, existirán correspondencias entre cada valor de un tipo y el correspondiente valor del otro tipo.

F.4.2 Para un tipo X1, creado a partir de otro tipo, X2, por medio de rotulación (véase la cláusula 30), se define la existencia de correspondencia de valores entre todos los miembros de X1 y los miembros correspondientes de X2.

NOTA – Si bien se define la existencia de correspondencias entre los valores de X1 y X2 en F.4.2, y entre los valores de X3 y X4 en F.4.3, cuando tales tipos estén incorporados en definiciones de tipo no idénticas, sino distintas (como SEQUENCE o CHOICE), las definiciones de tipo resultantes (los tipos SEQUENCE o CHOICE) no serán idénticas y no habrá correspondencias de valores entre ellas.

F.4.3 Para un tipo X3, creado seleccionando valores de cualquier tipo gobernador X4 por medio de la construcción de conjunto de elementos o de la subtipificación, se define la existencia de correspondencias de valores entre sus miembros y los miembros del tipo gobernante que fueron seleccionados por la construcción de conjunto de elementos o por la subtipificación. La presencia o ausencia de un marcador de extensión no afecta a esta regla.

F.4.4 En F.5 se especifican correspondencias de valores adicionales entre algunos de los tipos de cadenas de caracteres.

F.4.5 Se define la existencia de una correspondencia entre todos los valores de cualquier tipo definido como entero con valores nombrados y cualquier tipo entero definido sin valores nombrados, o con valores nombrados diferentes, o con nombres diferentes para los valores nombrados, o ambos.

NOTA – La existencia de correspondencias de valores no afecta a ningún requisito de la regla de ámbito de aplicación sobre la utilización de los nombres de los valores nombrados. Sólo pueden utilizarse en un ámbito gobernado por el tipo en el que se han definido, o por un nombre de referencia a ese tipo.

F.4.6 Se define la existencia de una correspondencia entre todos los valores de cualquier tipo definido como tipo de cadena de bits con bits nombrados y de cualquier tipo de cadena de bits definido sin bits nombrados, o con diferentes bits nombrados, con diferentes nombres para los bits nombrados, o ambos.

NOTA – La existencia de correspondencias de valores no afecta a ningún requisito de la regla de ámbito de aplicación sobre la utilización de los nombres de los valores nombrados. Sólo pueden utilizarse en un ámbito gobernado por el tipo en el que se han definido, o por un nombre de referencia a ese tipo.

F.5 Correspondencias de valores adicionales definidas para los tipos de cadenas de caracteres

F.5.1 Hay dos grupos de tipos de cadena de caracteres restringidos, el grupo A (véase F.5.2) y el grupo B (véase F.5.3). Se define la existencia de correspondencias de valores entre todos los tipos del grupo A, y las referencias a los valores de cada tipo pueden utilizarse cuando éste es gobernado por cualquiera de los otros tipos. En cuanto a los tipos del grupo B, no existen correspondencias de valores entre ellos, ni entre los tipos del grupo A y los tipos del grupo B.

F.5.2 El grupo A consiste en:

UTF8String
 NumericString
 PrintableString
 IA5String
 VisibleString (ISO646String)
 UniversalString
 BMPString

F.5.3 El grupo B consiste en:

TeletexString (T61String)
 VideotexString
 GraphicString
 GeneralString

F.5.4 Las correspondencias de valores del grupo A se especifican estableciendo las correspondencias entre los valores de cadena de caracteres de cada tipo y la UniversalString, y aplicando después la propiedad transitiva de las correspondencias de valores. Para establecer la correspondencia entre valores de uno de los tipos del grupo A y la UniversalString, se sustituye la cadena por una UniversalString de la misma longitud, asociando cada carácter como se especifica a continuación.

F.5.5 Formalmente, el conjunto de valores abstractos de UTF8String es el mismo que se produce en UniversalString pero con un rótulo diferente (véase 36.13), y cada valor abstracto de UTF8String se asocia al correspondiente valor abstracto de UniversalString.

F.5.6 Los caracteres gráficos utilizados para formar los tipos `NumericString` y `PrintableString` tienen correspondencias reconocibles y sin ambigüedad con un subconjunto de los caracteres gráficos asignados a los primeros 128 caracteres de ISO/CEI 10646. La correspondencia de estos tipos se define utilizando esta correspondencia de caracteres gráficos.

F.5.7 La correspondencia entre `IA5String` y `VisibleString` y la `UniversalString` se establece asociando cada carácter con el carácter de la `UniversalString` que tenga un valor idéntico (32 bits) en la codificación BER de `UniversalString` como el valor (8 bits) de la codificación BER de `IA5String` y `VisibleString`.

F.5.8 `BMPString` es formalmente un subconjunto de `UniversalString`, y los valores abstractos correspondientes tienen correspondencias de valores.

F.6 Requisitos específicos de compatibilidad de tipo y de valor

En la presente subcláusula se utiliza el concepto de correspondencia de valores para hacer precisiones respecto a la legalidad de ciertas construcciones ASN.1.

F.6.1 Toda ocurrencia de "Value", *x-notation* (notación *x*), con un tipo gobernante "Y" identifica el valor, *y-val* (valor *y*), de dicho tipo gobernante que tiene una correspondencia de valor con el valor *x-val* (valor *x*) especificado por la *x-notation*. Se requiere que dicho valor exista.

Considérese, por ejemplo, la "x" de la última línea de lo siguiente:

```
X ::= [0] INTEGER (0..30)
x X ::= 29
Y ::= [1] INTEGER (25..35)
Z1 ::= Y (x | 30)
```

Estas construcciones ASN.1 son legales, y en la última asignación la *x-notation* "x" hace referencia al *x-val* 29 de "X" y, a través de la correspondencia de valores, identifica el *y-val* 29 de "Y". La *x-notation* 30 hace referencia al *y-val* 30 de "Y", y "Z1" es el conjunto de valores 29 y 30. Por otra parte, la asignación:

```
Z2 ::= Y (x | 20)
```

es ilegal porque no existe un *y-val* al que pueda referirse la *x-notation* 20.

F.6.2 Siempre que aparezca "Type", *t-notation* (notación *t*), con un tipo gobernante "V" identifica al conjunto completo de valores de este tipo gobernante que tiene correspondencia de valor con cualquiera de los valores del "Type" *t-notation*. Este conjunto ha de contener al menos un valor.

Considérese por ejemplo, "W" de la última línea de lo siguiente:

```
V ::= [0] INTEGER (0..30)
W ::= [1] INTEGER (25..35)
Y ::= [2] INTEGER (31..35)
Z1 ::= V (W | 24)
```

"W" aporta los valores 25-30 a la aritmética del conjunto, con el resultado de que "Z1", tiene los valores 24-30. Por otra parte, la asignación:

```
Z2 ::= V (Y | 24)
```

es ilegal porque no hay valores en "Y" que se correspondan con los valores de "V".

F.6.3 Se requiere que el tipo de cualquier valor proporcionado como parámetro real tenga una correspondencia de valor con uno de los valores del tipo que gobierna el parámetro ficticio, y el valor que se identifica pertenece a ese tipo gobernante.

F.6.4 Si se proporciona un "Type" como parámetro real para un parámetro ficticio de un conjunto de valores, todos los valores de ese "Type" habrán de tener correspondencia con los valores del gobernante del parámetro ficticio del conjunto de valores. El parámetro real selecciona todo el conjunto de valores del gobernante que tienen correspondencia con el "Type".

F.6.5 La especificación del tipo, A, de un parámetro ficticio de valor o de conjunto de valores es ilegal a menos que todos los valores de A y todos los ejemplares de utilización de A en la parte derecha de la asignación puedan aplicarse legalmente en lugar del parámetro ficticio.

F.7 Ejemplos

F.7.1 En la presente subcláusula se dan ejemplos para ilustrar las subcláusulas F.3 y F.4.

F.7.2 Ejemplo 1

```

X ::= SEQUENCE
    {name VisibleString,
      age INTEGER}

X1 ::= SEQUENCE
    {name VisibleString,
      -- comment --
      age INTEGER}

X2 ::= [8] SEQUENCE
    {name VisibleString,
      age INTEGER}

X3 ::= SEQUENCE
    {name VisibleString,
      age AgeType}

AgeType ::= INTEGER

```

X, X1, X2 y X3 son definiciones de tipos idénticas. Las diferencias entre espacios en blanco y comentarios no son visibles, y la utilización de la referencia de tipo "AgeType" en X3 no afecta a la definición de tipo. Se señala, sin embargo, que si se modificara cualquiera de los identificadores de los elementos de la secuencia, los tipos dejarían de ser definiciones idénticas, y no habría correspondencia de valores entre ellos.

F.7.3 Ejemplo 2

```

B ::= SET
    {name VisibleString,
      age INTEGER}

B1 ::= SET
    {age INTEGER,
      name VisibleString}

```

estas definiciones son definiciones de tipo idénticas, con tal de que ninguna de ellas esté en un módulo con "AUTOMATIC TAGS" en su encabezamiento; de no ser así, no serán definiciones de tipo idénticas, y no existirá correspondencia de valores entre ellas. Pueden obtenerse ejemplos similares utilizando CHOICE y ENUMERATED (con la forma "identifier" de "EnumerationItem").

E.7.4 Ejemplo 3

```

C ::= SET
    {name [0]VisibleString,
      age INTEGER}

C1 ::= SET
    {name VisibleString,
      age INTEGER (1..64)}

```

estas definiciones no son definiciones de tipo idénticas entre sí, ni son idénticas tampoco a las de B o B1, y no hay correspondencia entre los valores de C y de C1, ni entre los valores de ninguno de ellos y los valores de B o B1.

F.7.5 Ejemplo 4

```

x  INTEGER { y (2) } ::= 3
z  INTEGER ::= x

```

es legal, y asigna el valor 3 a z según la correspondencia de valores definida en F.4.5.

F.7.6 Ejemplo 5

```

b1 BIT STRING ::= '101'B
b2 BIT STRING {version1(0), version2(1), version3(2)} ::= b1

```

es legal, y asigna el valor {version1, version3} a b2.

F.7.7 Ejemplo 6

Con las definiciones de F.1.1, los elementos de SEQUENCE de la forma:

```
X DEFAULT y
```

son legales cuando X es A, B, C, D, E o F, o cualquiera de los textos a la derecha de las asignaciones de tipo a estos nombres, e y es a, b, c, d, e, o f, con las siguientes excepciones: **E DEFAULT y** es ilegal para a, b, c, d, f, y **C DEFAULT e** es ilegal, porque en estos casos no existe correspondencia de valores entre la referencia de valor que falta y el tipo al que se suple.

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Transmisiones de señales radiofónicas, de televisión y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación