



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.680

Amendement 2
(06/99)

SÉRIE X: RÉSEAUX POUR DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Réseautage OSI et aspects systèmes – Notation de
syntaxe abstraite numéro un (ASN.1)

Technologies de l'information – Notation de syntaxe
abstraite numéro un: spécification de la notation de
base

Amendement 2: Modèle sémantique d'ASN.1

Recommandation UIT-T X.680 – Amendement 2

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE X
RÉSEAUX POUR DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

RÉSEAUX PUBLICS POUR DONNÉES	
Services et fonctionnalités	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalisation et commutation	X.50–X.89
Aspects réseau	X.90–X.149
Maintenance	X.150–X.179
Dispositions administratives	X.180–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200–X.209
Définitions des services	X.210–X.219
Spécifications des protocoles en mode connexion	X.220–X.229
Spécifications des protocoles en mode sans connexion	X.230–X.239
Formulaires PICS	X.240–X.259
Identification des protocoles	X.260–X.269
Protocoles de sécurité	X.270–X.279
Objets gérés des couches	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion et fonctions ODMA	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999

NORME INTERNATIONALE 8824-1
RECOMMANDATION UIT-T X.680

TECHNOLOGIES DE L'INFORMATION –
NOTATION DE SYNTAXE ABSTRAITE NUMÉRO UN:
SPÉCIFICATION DE LA NOTATION DE BASE

AMENDEMENT 2
Modèle sémantique d'ASN.1

Résumé

L'Amendement 2 à la Rec. UIT-T X.680 | ISO/CEI 8824-1 s'adresse principalement aux concepteurs d'outils. Il a pour objet de leur permettre d'interpréter la norme ASN.1 de la même façon en ce qui concerne les correspondances entre valeurs de différents types. Il indique:

- a) ce qu'est une définition ASN.1 légale et non légale;
- b) la valeur précise identifiée par un nom de référence de valeur;
- c) l'ensemble de valeurs précis identifié par un nom de référence d'ensemble de valeurs ou par un nom de référence de type.

Source

L'Amendement 2 à la Recommandation X.680 de l'UIT-T a été approuvé le 18 juin 1999. Un texte identique est publié comme Norme internationale ISO/CEI 8824-1.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, le terme *exploitation reconnue (ER)* désigne tout particulier, toute entreprise, toute société ou tout organisme public qui exploite un service de correspondance publique. Les termes *Administration*, *ER* et *correspondance publique* sont définis dans la *Constitution de l'UIT (Genève, 1992)*.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 1999

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

	<i>Page</i>
1) Paragraphe 3.8	1
2) Nouveau paragraphe 5.9	1
5.9 Références de valeurs et typage de valeurs	1
3) Nouveau paragraphe 13.6	2
4) Nouveau paragraphe 13.7	2
5) Paragraphe 15.2	2
6) Paragraphe 36.7	2
7) Paragraphe 48.3.2	2
8) Paragraphe 48.5.2	2
9) Paragraphe 48.8.2	2
10) Nouvelle Annexe F	3
Annexe F – Règles applicables à la compatibilité de types et de valeurs	3
F.1 Nécessité du concept de correspondance entre valeurs (introduction didactique)	3
F.2 Correspondances entre valeurs	6
F.3 Définition de types identiques	6
F.4 Spécification des correspondances entre valeurs	8
F.5 Correspondances supplémentaires définies entre valeurs des types de chaînes de caractères	9
F.6 Conditions particulières de la compatibilité de types et de valeurs	9
F.7 Exemples	10

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION –
NOTATION DE SYNTAXE ABSTRAITE NUMÉRO UN:
SPÉCIFICATION DE LA NOTATION DE BASE**

**AMENDEMENT 2
Modèle sémantique d'ASN.1**

1) Paragraphe 3.8

*Ajouter les définitions 3.8.39 bis et 3.8.71 bis au paragraphe 3.8 et remplacer la définition de **gouvernant; gouverneur** par le nouveau paragraphe 3.8.39 ci-dessous:*

3.8.39 gouvernant; gouverneur (type): référence ou définition de type qui affecte l'interprétation d'une partie de la syntaxe ASN.1, exigeant qu'elle fasse référence à des valeurs du type gouvernant.

3.8.39 bis définitions de types identiques: deux instances de la production "Type" ASN.1 (voir article 16) sont définies comme des définitions de types identiques si, après application des transformations spécifiées dans l'Annexe F, elles sont constituées des listes ordonnées identiques d'unités lexicales ASN.1 (voir article 11).

3.8.71 bis correspondance entre valeurs: relation biunivoque entre des valeurs de deux types, qui permet d'utiliser une référence à une valeur d'un type comme référence à une valeur de l'autre type. On peut l'utiliser, par exemple, pour spécifier des sous-types et des valeurs par défaut (voir Annexe F).

Ajouter au paragraphe 3.8.50:

{texte existant}, et qui gouverne la notation de sous-type.

2) Nouveau paragraphe 5.9

Ajouter un nouveau paragraphe 5.9 comme suit:

5.9 Références de valeurs et typage de valeurs

5.9.1 L'ASN.1 décrit une notation de définition de valeur qui permet de donner un nom à une valeur d'un type spécifié. Ce nom peut être utilisé chaque fois qu'il est nécessaire de faire référence à cette valeur. L'Annexe F décrit et spécifie le mécanisme de **correspondance entre valeurs** qui permet à un nom de référence de valeur affecté à une valeur d'un type d'identifier une valeur d'un second type. Ainsi, on peut utiliser une référence à une valeur du premier type lorsqu'une référence à une valeur du second type est nécessaire.

5.9.2 Dans le corps des normes ASN.1, un texte définit la légalité (ou non) des constructions mettant en jeu plusieurs types, mais qui doivent être **compatibles**. Par exemple, le type utilisé pour définir une référence de valeur doit être **compatible** avec le type qui gouverne l'utilisation de cette référence de valeur. L'Annexe F normative utilise le concept de correspondance entre valeurs pour donner une indication précise sur la légalité ou non d'une construction ASN.1 donnée.

3) Nouveau paragraphe 13.6

Ajouter un nouveau paragraphe 13.6 comme suit:

13.6 Lorsqu'un type "DefinedType" est utilisé dans une notation gouvernée par un "Type" (par exemple un "SubtypeElementSpec"), le type "DefinedType" doit être compatible avec le "Type" gouvernant, comme spécifié au paragraphe F.6.2.

4) Nouveau paragraphe 13.7

Ajouter un nouveau paragraphe 13.7 comme suit:

13.7 Chaque occurrence d'une valeur "DefinedValue" dans une spécification ASN.1 est gouvernée par un "Type" et cette valeur "DefinedValue" doit faire référence à une valeur d'un type compatible avec le "Type" gouvernant, comme spécifié au paragraphe F.6.2.

5) Paragraphe 15.2

Au paragraphe 15.2, modifier la phrase suivant le formalisme BNF en insérant immédiatement avant "sera" le texte suivant:

est gouvernée par le "Type" et

6) Paragraphe 36.7

Ajouter à l'exemple de la Note 3 du paragraphe 36.7 ce qui suit:

Autre définition non ambiguë de "mystring":

```
mystring MyAlphabet(BasicLatin) ::= "POINT"
```

Formellement, "mystring" est une référence de valeur à une valeur d'un sous-ensemble de "MyAlphabet", mais elle peut, en vertu des règles de correspondance entre valeurs de l'Annexe F, être utilisée pour référencer la même chaîne mais en tant que valeur du type "MyAlphabet".

7) Paragraphe 48.3.2

Remplacer le paragraphe 48.3.2 par le texte suivant:

48.3.2 La notation "ContainedSubtype" spécifie toutes les valeurs du type parent qui sont aussi des valeurs du "Type", qui doit lui-même être compatible avec le type parent, comme spécifié au paragraphe F.6.3

8) Paragraphe 48.5.2

Au paragraphe 48.5.2, supprimer le passage suivant: ", ou aux types obtenus à partir de ces derniers par étiquetage".

9) Paragraphe 48.8.2

Au paragraphe 48.8.2, supprimer le passage suivant: ", ainsi qu'aux types qui en dérivent par étiquetage".

10) Nouvelle Annexe F

Ajouter une nouvelle Annexe F comme suit:

Annexe F

Règles applicables à la compatibilité de types et de valeurs

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

La présente annexe s'adresse principalement aux concepteurs d'outils en vue de leur permettre d'interpréter la notation ASN.1 de la même façon. Elle a pour objet de spécifier clairement ce qu'est une définition ASN.1 légale et non légale, et d'indiquer la valeur précise identifiée par un nom de référence de valeur ainsi que l'ensemble de valeurs précis identifié par un nom de référence d'ensemble de valeurs ou par un nom de référence de type. Elle ne vise pas à donner une définition des transformations valides des définitions ASN.1 à d'autres fins que celles indiquées ci-dessus.

F.1 Nécessité du concept de correspondance entre valeurs (introduction didactique)

F.1.1 Considérons les définitions ASN.1 suivantes:

```
A ::= INTEGER
B ::= [1] INTEGER
C ::= [2] INTEGER (0..6, ...)
D ::= [2] INTEGER (0..6, ..., 7)
E ::= INTEGER (7..20)
F ::= INTEGER {rouge(0), blanc(1), bleu(2), vert(3), violet(4)}
a A ::= 3
b B ::= 4
c C ::= 5
d D ::= 6
e E ::= 7
f F ::= vert
```

F.1.2 Il est clair que les références de valeurs "a", "b", "c", "d", "e" et "f" peuvent être utilisées dans la notation de valeur gouvernée respectivement par "A", "B", "C", "D", "E" et "F". Par exemple:

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

et:

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

et:

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

sont toutes des expressions valables compte tenu des définitions données au paragraphe F.1.1. Toutefois, si le type "A" ci-dessus était remplacé par "B", "C", "D", "E" ou "F", les déclarations qui en découlent seraient-elles illégales? De même, si la référence de valeur "a" ci-dessus était remplacée dans chacun de ces cas par "b", "c", "d", "e" ou "f", les déclarations qui en découlent seraient-elles légales?

F.1.3 Il serait plus délicat d'envisager dans chaque cas de remplacer la référence de type par le texte explicite à droite de sa définition, par exemple:

```
f INTEGER {rouge(0), blanc(1), bleu(2), vert(3), violet(4)} ::= vert
W ::= SEQUENCE {
    w1 INTEGER {rouge(0), blanc(1), bleu(2), vert(3), violet(4)}
    DEFAULT f}
x INTEGER {rouge(0), blanc(1), bleu(2), vert(3), violet(4)} ::= f
Y ::= INTEGER {rouge(0), blanc(1), bleu(2), vert(3), violet(4)}(1..f)
```

La notation ci-dessus serait-elle une notation ASN.1 légale?

F.1.4 Les utilisateurs seraient mal avisés de rédiger un texte semblable à certains des exemples donnés ci-dessus, même si la plupart sont légaux (voir le dernier exemple), car ces exemples sont pour le moins obscurs, voire au pire confus. Toutefois, on utilise souvent une référence de valeur d'un type (pas nécessairement un type "INTEGER") comme valeur par défaut pour un type gouverneur déduit du premier par étiquetage ou sous-typage. Le concept de **correspondance entre valeurs** est introduit en vue de donner un moyen clair et précis de déterminer les constructions qui, comme celles citées ci-dessus, sont légales.

F.1.5 Examinons de nouveau:

```
C ::= [2] INTEGER (0..6, ...)
```

```
E ::= INTEGER (7..20)
```

```
F ::= INTEGER {rouge(0), blanc(1), bleu(2), vert(3), violet(4)}
```

Dans chaque cas, un nouveau type est créé. Pour le type "F", nous pouvons clairement identifier une correspondance biunivoque entre les valeurs qu'il contient et les valeurs du type prédéfini "INTEGER". Pour les types "C" et "E", nous pouvons identifier clairement une correspondance biunivoque entre les valeurs qu'ils contiennent et un sous-ensemble de valeurs du type prédéfini "INTEGER". Nous appelons cette relation une **correspondance entre valeurs** des deux types. En outre, étant donné que les valeurs des types "F", "C" et "E" sont toutes en correspondance biunivoque avec les valeurs de sous-ensembles du type "INTEGER", nous pouvons utiliser ces correspondances pour définir des correspondances entre les valeurs des types "F", "C" et "E". La Figure F.1 illustre cet exemple pour les types "F" et "C".

F.1.6 Lorsque nous avons une référence à une valeur du type "C":

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

qui, dans un certain contexte doit être utilisée pour identifier une valeur du type "F", s'il existe une correspondance entre cette valeur du type "C" et une (seule) valeur du type "F", nous pouvons définir (et nous définissons) "c" comme une référence légale à une valeur du type "F". Ce cas est illustré par la Figure F.2, où la référence de valeur "c" sert à identifier une valeur du type "F", et peut être utilisée à la place d'une référence directe "f1" qu'il serait sans cela nécessaire de définir:

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

F.1.7 Il convient de noter que dans certains cas, des valeurs d'un type (les valeurs 7 à 20 du type "A" du paragraphe F.1.1 par exemple) seront en correspondance avec des valeurs d'un autre type (les valeurs 7 à 20 du type "E" du paragraphe F.1.1 par exemple), mais que d'autres valeurs du même type (à partir de la valeur 21 du type "A") ne le seront pas. Une référence à ces valeurs du type "A" ne serait pas une référence valable à une valeur du type "E". (Dans cet exemple, l'ensemble des valeurs du type "E" est en correspondance avec les valeurs d'un sous-ensemble de l'ensemble des valeurs du type "A". En règle générale, il peut y avoir un sous-ensemble de valeurs du premier type, pour lesquelles il existe une correspondance avec les valeurs d'un sous-ensemble du second type, les valeurs appartenant à ces sous-ensembles n'étant pas en correspondance.)

F.1.8 Dans le corps des normes ASN.1, un texte est utilisé pour spécifier la légalité concernant les cas présentés ci-dessus ou des cas similaires. Le paragraphe F.6 indique les conditions précises de légalité et il convient de s'y référer en cas de doute au sujet d'une construction complexe.

NOTE – Le fait qu'une correspondance entre valeurs soit définie entre deux occurrences de constructions de "Type" permet d'utiliser des références de valeur établies à l'aide d'une construction de "Type" pour identifier des valeurs dans une autre construction de "Type" suffisamment semblable. Cela permet de typer les paramètres fictifs et réels en utilisant deux constructions de "Type" textuelles distinctes sans enfreindre les règles de compatibilité des paramètres fictifs et réels. Cela permet aussi de spécifier les champs de classes d'objets informationnels en utilisant une construction de "Type" et la valeur correspondante dans la définition d'un objet informationnel à l'aide d'une construction de "Type" distincte suffisamment similaire. (La liste de ces exemples n'est pas exhaustive.) Il est toutefois recommandé de n'avoir recours à ces possibilités que pour des cas simples comme "SEQUENCE OF INTEGER", ou "CHOICE {int INTEGER, id OBJECT IDENTIFIER}", et non pour des constructions de "Type" plus complexes.

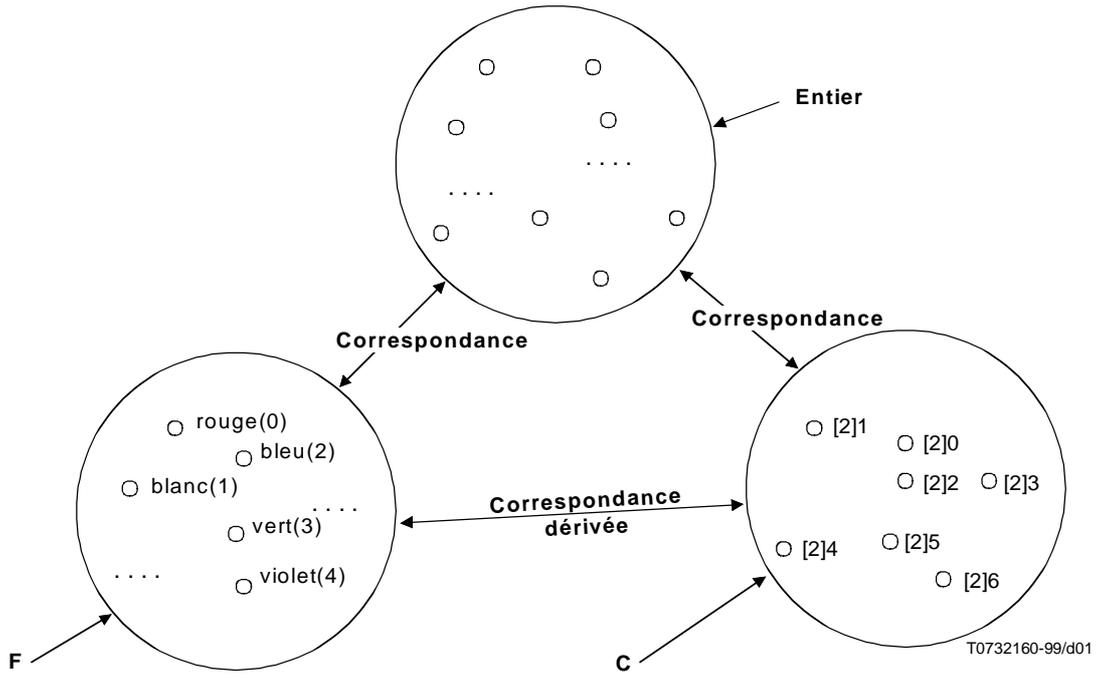


Figure F.1

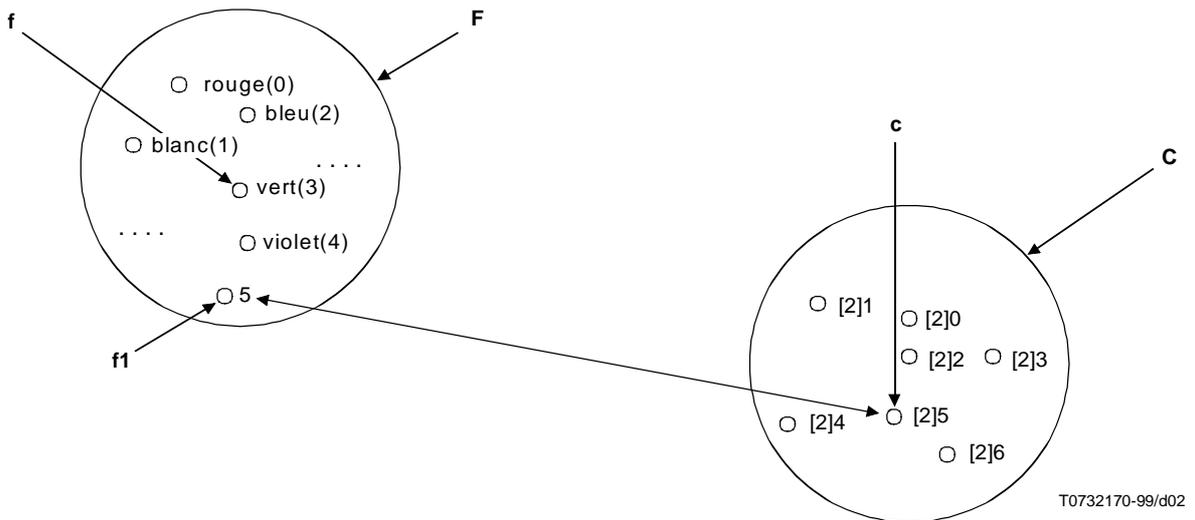


Figure F.2

F.2 Correspondances entre valeurs

F.2.1 Le modèle sous-jacent est constitué de types, considérés comme des conteneurs qui ne se chevauchent pas et qui contiennent des valeurs, chaque occurrence de la construction de "Type" ASN.1 définissant un nouveau type distinct (voir Figures F.1 et F.2). La présente annexe indique dans quelles conditions des **correspondances** existent entre les valeurs de ces types, ce qui permet d'utiliser une référence de valeur d'un type lorsqu'une référence de valeur à un autre type est nécessaire.

Exemple:

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

"X" et "Y" sont des noms de référence (pointeurs) vers deux types distincts, mais une correspondance existe entre les valeurs de ces types, de sorte que l'on peut utiliser n'importe quelle référence à une valeur de "X" pour une valeur gouvernée par "Y" (par exemple, après le mot-clé DEFAULT).

F.2.2 Dans l'ensemble de toutes les valeurs possibles de la notation ASN.1, une correspondance entre valeurs se rapporte à une paire de valeurs. Une correspondance entre valeurs est une relation mathématique qui a les propriétés suivantes: elle est réflexive (chaque valeur ASN.1 est en correspondance avec elle-même), symétrique (si une valeur "x1" est en correspondance avec une valeur "x2", la valeur "x2" est automatiquement en correspondance avec la valeur "x1") et transitive (s'il y a, d'une part, une correspondance entre une valeur "x1" et une valeur "x2", d'autre part, une correspondance entre une valeur "x2" et une valeur "x3", il existe automatiquement une correspondance entre "x1" et "x3").

F.2.3 En outre, étant donné deux types "X1" et "X2", considérés comme des ensembles de valeurs, la correspondance entre des valeurs dans "X1" et des valeurs dans "X2" est une relation univoque, c'est-à-dire que pour tout couple de valeurs "x1" de "X1", et "x2" de "X2", s'il y a une correspondance entre "x1" et "x2", alors:

- a) il n'y a pas de correspondance entre "x1" et une autre valeur de "X2" différente de "x2";
- b) il n'y a pas de correspondance entre une valeur quelconque de "X1" (autre que "x1") et "x2".

F.2.4 Lorsqu'il y a une correspondance entre une valeur "x1" et une valeur "x2", une référence à l'une de ces valeurs peut automatiquement servir de référence à l'autre si le type gouverneur l'exige.

NOTE – Le fait que des correspondances entre valeurs soient définies dans certaines constructions de "Type" vise simplement à accorder une plus grande souplesse dans l'utilisation de la notation ASN.1. **L'existence de ces correspondances entre valeurs n'implique en aucune façon que les deux types acheminent les mêmes sémantiques du point de vue de l'application**, mais il est **recommandé** de n'utiliser des constructions ASN.1 qui seraient illégales sans correspondances entre valeurs, que si les types correspondants ont réellement la même sémantique du point de vue de l'application. A noter que l'on trouvera souvent, dans des spécifications comportant de nombreuses définitions, des correspondances entre valeurs de types qui sont des constructions ASN.1 identiques, mais qui ont une sémantique totalement différente du point de vue de l'application; toutefois, ces correspondances entre valeurs ne sont jamais utilisées pour déterminer la légalité de l'ensemble de la spécification.

F.3 Définition de types identiques

F.3.1 Le concept de **définition de types identiques** permet de définir des correspondances entre valeurs entre deux instances de "Type" identiques ou suffisamment similaires pour être normalement interchangeables. Afin de préciser la signification de "suffisamment similaires", le présent paragraphe indique une série de transformations appliquées à chacune des instances de "Type" en vue de produire une **forme normale** pour ces instances de "Type". Les deux instances de "Type" sont définies comme étant des définitions de types identiques si et seulement si leurs formes normales sont des listes ordonnées identiques des mêmes unités lexicales ASN.1 (voir article 11).

F.3.2 Chaque occurrence de "Type" dans une spécification ASN.1 est une liste ordonnée d'unités lexicales définies à l'article 11. La forme normale est obtenue par application des transformations définies du paragraphe F.3.2.1 au paragraphe F.3.2.4 (dans cet ordre).

F.3.2.1 Toutes les unités lexicales commentaires (voir paragraphe 11.6) doivent être supprimées.

F.3.2.2 Les transformations suivantes ne sont pas récurrentes et ne doivent donc être appliquées qu'une seule fois, dans un ordre quelconque:

- a) pour chaque type "integer": la liste "NamedNumberList" (voir paragraphe 18.1) est réordonnée de façon que les identificateurs ("identifiers") soient dans l'ordre alphabétique ("a" en premier, "z" en dernier);
- b) pour chaque type "enumerated": des numéros sont ajoutés, comme spécifié au paragraphe 19.3, à toute unité lexicale "EnumerationItem" (voir paragraphe 19.1) qui est un "identifiant" (sans numéro); puis l'énumération "RootEnumeration" est réordonnée de façon que les identificateurs ("identifiers") soient dans l'ordre alphabétique ("a" en premier, "z" en dernier);
- c) pour chaque type "bitstring": la liste "NamedBitList" (voir paragraphe 21.1) est, si elle existe, réordonnée de façon que les identificateurs ("identifiers") soient dans l'ordre alphabétique ("a" en premier, "z" en dernier);
- d) pour chaque valeur d'identificateur d'objet: chaque composant "ObjectIdComponent" est transformé en sa forme "NumberForm" correspondante, conformément à la sémantique de l'article 31 (voir l'exemple au paragraphe 31.11);
- e) pour les types "sequence" (voir article 24) et "set" (voir article 26): toute extension de la forme "ExtensionAndException", "ExtensionAdditions" est coupée et collée à la fin de la production "ComponentTypeLists"; la production "OptionalExtensionMarker" est supprimée si elle est présente.

Si l'option "TagDefault" est égale à "IMPLICIT TAGS", le mot clé "IMPLICIT" est ajouté à toutes les instances de "Tag" (voir article 30) sauf dans les cas suivants:

- le mot clé "IMPLICIT" est déjà présent;
- le mot réservé "EXPLICIT" est présent;
- le type étiqueté est un type "CHOICE";
- il s'agit d'un type ouvert.

Si l'option "TagDefault" est égale à "AUTOMATIC TAGS", la décision d'appliquer l'étiquetage automatique est prise conformément au paragraphe 24.2 (l'étiquetage automatique sera effectué ultérieurement).

NOTE – Les paragraphes 24.3 et 26.2 indiquent que la présence d'une unité lexicale "Tag" dans une notation "ComponentType" qui a été insérée à la suite du remplacement de "Components of Type" n'empêche pas la transformation d'étiquetage automatique.

Si l'option "ExtensionDefault" est égale à "EXTENSIBILITY IMPLIED", des points de suspension ("...") sont ajoutés après la notation "ComponentTypeLists", s'ils ne sont pas déjà présents;

- f) pour le type "choice" (voir article 28): la liste "RootAlternativeType" est réordonnée de façon que les identificateurs des types "NameType" soient dans l'ordre alphabétique ("a" en premier, "z" en dernier). Si l'option "TagDefault" est égale à "IMPLICIT TAGS", le mot clé "IMPLICIT" est ajouté à toutes les instances de "Tag" (voir article 30) sauf dans les cas suivants:
 - le mot clé "IMPLICIT" est présent;
 - le mot réservé "EXPLICIT" est présent;
 - le type étiqueté est un type "CHOICE";
 - il s'agit d'un type ouvert.

Si l'option "TagDefault" est égale à "AUTOMATIC TAGS", la décision d'appliquer l'étiquetage automatique est prise conformément au paragraphe 28.3 (l'étiquetage automatique sera effectué ultérieurement). Si l'option "ExtensionDefault" est égale à "EXTENSIBILITY IMPLIED", des points de suspension ("...") sont ajoutés après la notation "AlternativeTypeLists", s'ils ne sont pas déjà présents.

F.3.2.3 Les transformations suivantes doivent faire l'objet d'une application récursive dans l'ordre spécifié, jusqu'à ce qu'un point de repère soit atteint:

- a) pour chaque valeur d'identificateur d'objet (voir paragraphe 31.3): si la définition de valeur commence par une valeur définie ("DefinedValue"), celle-ci est remplacée par sa définition;
- b) pour les types "sequence" et "set": toutes les instances de l'option "COMPONENT OF types" (voir article 24) sont transformées conformément aux articles 24 et 26;
- c) pour les types "sequence", "set" et "choice": s'il a été antérieurement décidé d'effectuer un étiquetage automatique [voir les points e) et f) du paragraphe F.3.2.2], l'étiquetage automatique est appliqué conformément aux articles 24, 26 et 28;
- d) pour le type "selection": la construction est remplacée par l'option choisie conformément à l'article 29;

- e) toutes les références de type sont remplacées par leurs définitions conformément aux règles suivantes:
- si le type remplaçant est une référence au type qui subit la transformation, la référence de type est remplacée par une unité lexicale spéciale ne correspondant à aucune autre unité lexicale qu'elle-même;
 - si le type remplaçant est de type "séquence de" ou "ensemble de", les contraintes faisant éventuellement suite au type remplacé sont transférées devant le mot clé "OF";
 - si le type remplacé est de type valeur paramétrée ou ensemble de valeurs paramétrées (voir 8.2 de la Rec. UIT-T X.683 | ISO/CEI 8824-4), chaque unité lexicale "DummyReference" est remplacée par l'élément "ActualParameter" correspondant;
- f) toutes les références de valeur sont remplacées par leurs définitions, si la valeur remplacée est une valeur paramétrée (voir 28.3 de la Rec. UIT-T X.683 | ISO/CEI 8824-4, chaque valeur lexicale "DummyReference" sera remplacée par l'élément "ActualParameter" correspondant.

NOTE – Avant de remplacer une quelconque référence de valeur, les procédures de la présente annexe doivent être appliquées afin de s'assurer que cette référence de valeur désigne, directement ou par mappage de valeur, une valeur de son type directeur.

F.3.2.4 Pour le type "set": la liste "RootComponentTypeList" est réordonnée de façon que les types "ComponentType" soient dans l'ordre alphabétique ("a" en premier, "z" en dernier).

NOTE – Dans les définitions données aux paragraphes 11.9 ("bstring"), 11.10 ("hstring") et 11.11 ("cstring"), il est indiqué que les blancs et les retours à la ligne ne sont pas importants dans ces unités lexicales. Si deux occurrences de ces unités lexicales contiennent différentes utilisations des blancs et des retours à la ligne, elles sont traitées comme des unités lexicales identiques aux fins des dispositions du paragraphe F.3.3.

F.3.3 Si deux instances de "Type", transformées en forme normale, sont des listes identiques d'unités lexicales ASN.1 (voir article 11), ce sont des définitions de types identiques sauf dans le cas suivant: si une notation "objectclassreference" (voir 7.1 de la Rec. UIT-T X.681 | ISO/CEI 8824-2) apparaît dans la forme normalisée du "Type", les deux types **ne** sont **pas** considérés comme étant des définitions de types identiques, et il n'y aura pas de correspondance entre leurs valeurs (voir paragraphe F.4 ci-dessous).

NOTE – Cette exception a été ajoutée pour éviter de devoir fournir des règles de transformation en forme normale pour les éléments de syntaxe concernant les classes d'objets informationnels, les objets informationnels et la notation d'ensemble d'objets informationnels. De même, la spécification pour la normalisation des notations de valeurs et d'ensembles de valeurs n'a pas été insérée pour le moment. Si cette spécification s'avère nécessaire, elle pourra être fournie dans une future version de la présente Recommandation | Norme internationale. Le concept de définition de types identiques et de correspondances entre valeurs a été introduit pour faire en sorte que l'on puisse utiliser des constructions ASN.1 simples soit en ayant recours au nom de référence, soit en recopiant le texte. Il n'a pas été jugé utile de fournir cette fonctionnalité pour des instances plus complexes de "Type" comprenant des classes d'objets informationnels, etc.

F.4 Spécification des correspondances entre valeurs

F.4.1 Si deux occurrences de "Type" sont des définitions de types identiques, conformément aux règles du paragraphe F.3, il existe, pour chaque valeur de l'un des types, une correspondance avec une valeur de l'autre type.

F.4.2 Pour un type "X1" créé à partir d'un type quelconque "X2" par étiquetage (voir article 30), il existe pour toute valeur de "X1" une correspondance avec une valeur de "X2".

NOTE – Bien que des correspondances entre valeurs soient définies entre les valeurs de "X1" et de "X2" dans le présent F.4.2, et entre les valeurs de "X3" et de "X4" au paragraphe F.4.3, si ces types sont utilisés pour définir d'autres types (comme "SEQUENCE" ou "CHOICE") qui ne diffèrent que par le fait que le premier utilise "X1" ou "X3" et le second "X2" (ou "X4"), les définitions de ces types ("SEQUENCE" ou "CHOICE") ne seront pas des définitions de types identiques et il n'y aura pas de correspondance entre leurs valeurs.

F.4.3 Pour un type "X3" créé en sélectionnant des valeurs conformes à un type gouverneur quelconque "X4" par la définition d'un ensemble de valeurs ou par sous-typage, une correspondance existe entre les valeurs du nouveau type "X3" et les valeurs du type "X4" qui ont été choisies par la contrainte de sous-typage ou par l'ensemble de valeurs. La présence ou l'absence d'un marqueur d'extension n'a aucune incidence sur cette règle.

F.4.4 Le paragraphe F.5 définit des correspondances supplémentaires entre valeurs relatives à certains types de chaînes de caractères.

F.4.5 Une correspondance est définie entre toutes les valeurs d'un type entier défini avec des valeurs nommées et un type entier défini sans valeurs nommées, ou avec des valeurs nommées différentes, ou encore avec des valeurs nommées différemment, ou les deux à la fois.

NOTE – L'existence de cette correspondance entre valeurs n'a aucune incidence sur les exigences des règles de visibilité quant à l'utilisation des noms des valeurs nommées. On ne peut les utiliser que dans le domaine de visibilité du type associé, ou en utilisant une référence à ce type.

F.4.6 Une correspondance entre valeurs est définie entre toutes les valeurs d'un type chaîne de bits définie avec des bits nommés et un type chaîne de bits définie sans bits nommés, ou avec des bits nommés différents, ou encore avec des bits nommés différemment, ou les deux à la fois.

NOTE – L'existence de cette correspondance entre valeurs n'a aucune incidence sur les exigences des règles de visibilité quant à l'utilisation des noms des bits nommés. On ne peut les utiliser que dans le domaine de visibilité du type associé, ou en utilisant une référence à ce type.

F.5 Correspondances supplémentaires définies entre valeurs des types de chaînes de caractères

F.5.1 Il y a deux groupes de types de chaînes de caractères à l'alphabet restreint: le groupe A (voir paragraphe F.5.2) et le groupe B (voir paragraphe F.5.3). Des correspondances entre valeurs sont définies entre tous les types du groupe A et les références de valeurs de ces types peuvent être utilisées lorsqu'elles sont gouvernées par l'un des autres types. Il n'y a jamais de correspondance entre les valeurs des types du groupe B, ni entre les valeurs d'un type du groupe A et les valeurs d'un type du groupe B.

F.5.2 Le groupe A comprend:

UTF8String
 NumericString
 PrintableString
 IA5String
 VisibleString (ISO646String)
 UniversalString
 BMPString

F.5.3 Le groupe B comprend:

TeletexString (T61String)
 VideotexString
 GraphicString
 GeneralString

F.5.4 Pour spécifier les correspondances entre valeurs dans le groupe A, on utilise la correspondance entre les valeurs de chaînes de caractères de chacun des types et les valeurs du type "UniversalString", puis on utilise la transitivité de la correspondance entre valeurs. La correspondance entre les valeurs de l'un des types du groupe A et les valeurs du type "UniversalString", est obtenue par remplacement de la chaîne par une chaîne "UniversalString" de même longueur, chaque caractère étant remplacé par le caractère correspondant comme indiqué ci-dessous.

F.5.5 Formellement, l'ensemble des valeurs abstraites du type "UTF8String" est le même que celui du type "UniversalString" mais avec une étiquette différente (voir paragraphe 36.13) et chaque valeur abstraite du type "UTF8String" est définie de sorte qu'il y ait correspondance avec la valeur abstraite du type "UniversalString" qui lui est associée.

F.5.6 Les glyphes (formes de caractères imprimées) pour les caractères servant à former les types "NumericString" et "PrintableString" sont en correspondance d'une manière reconnaissable et non ambiguë avec un sous-ensemble des glyphes attribués aux premiers 128 caractères de la Norme ISO/CEI 10646. La correspondance entre les valeurs de ces types est définie à l'aide de cette correspondance entre glyphes.

F.5.7 Les chaînes "IA5String" et "VisibleString" sont en correspondance avec "UniversalString", chaque caractère du type "IA5String" ou "VisibleString" est associé au caractère du type "UniversalString" qui a la même valeur (32 bits) dans le codage BER de "UniversalString" que la valeur (8 bits) du codage BER de "IA5String" ou "VisibleString".

F.5.8 La chaîne "BMPString" est formellement un sous-ensemble de la chaîne "UniversalString" et les valeurs abstraites de ces deux types sont donc en correspondance.

F.6 Conditions particulières de la compatibilité de types et de valeurs

Dans le présent paragraphe, on utilise le concept de correspondance entre valeurs pour fournir un texte précis pour la légalité de certaines constructions ASN.1.

F.6.1 Toute occurrence de "Value", *x-notation*, utilisée avec un type gouverneur, "Y", identifie la valeur, *y-val*, du type gouverneur "Y" qui est en correspondance avec la valeur *x-val* spécifiée par *x-notation*. Toutefois, il faut que cette valeur existe.

Par exemple, considérons l'occurrence de "x" dans la dernière ligne des expressions:

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

Ces constructions ASN.1 sont légales et dans la dernière définition, la référence *x-notation* "x" fait référence à *x-val* 29 du type "X" et, grâce à la correspondance entre valeurs, identifie *y-val* 29 dans "Y". La référence *x-notation* 30 fait référence à *y-val* 30 de type "Y", et "Z1" est l'ensemble des valeurs 29 et 30. Par ailleurs, la définition:

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

est illégale car il n'y a pas de valeur *y-val* à laquelle la référence *x-notation* 20 puisse être associée.

F.6.2 Toute occurrence de "Type", *t-notation*, qui a un type gouverneur "V", identifie l'ensemble complet de valeurs du type gouverneur "V" pour lesquelles il existe une correspondance avec l'une quelconque des valeurs dans la *t-notation* "Type". Cet ensemble doit contenir au moins une valeur.

Par exemple, considérons l'occurrence de "W" dans la dernière ligne des expressions suivantes:

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

"W" fournissant les valeurs de "V" de 25 à 30, "Z1" a les valeurs 24 à 30. Par ailleurs, la définition:

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

est illégale car il n'y a pas de valeur du type "Y" qui soit en correspondance avec une valeur de type "V".

F.6.3 Le type de toute valeur fournie comme paramètre réel doit être tel qu'il y ait une correspondance entre cette valeur et l'une des valeurs du type qui gouverne le paramètre fictif, et c'est une valeur de ce type gouverneur qui sera effectivement utilisée.

F.6.4 Si un "Type" est fourni comme paramètre réel quand le paramètre fictif désigne un ensemble de valeurs, toutes les valeurs de ce "Type" doivent être en correspondance avec des valeurs du type qui gouverne le paramètre fictif. Le paramètre réel effectif sera l'ensemble des valeurs du type indiqué par le gouverneur du paramètre fictif qui sont en correspondance avec les valeurs de "Type".

F.6.5 L'utilisation de valeurs d'un type "A", à la place de valeurs d'un paramètre fictif qui désigne une valeur ou un ensemble de valeurs, est illégale sauf si pour toutes les valeurs de "A", et pour chaque utilisation de "A" dans le membre de droite de la définition, cette valeur de "A" peut être appliquée légalement à la place du paramètre fictif.

F.7 Exemples

F.7.1 Le présent paragraphe donne des exemples pour illustrer les dispositions des paragraphes F.3 et F.4.

F.7.2 Exemple 1

```
X ::= SEQUENCE
    {nom VisibleString,
     age INTEGER}
X1 ::= SEQUENCE
    {nom VisibleString,
     -- commentaire --
     age INTEGER}
X2 ::= [8] SEQUENCE
    {nom VisibleString,
     age INTEGER}
X3 ::= SEQUENCE
    {nom VisibleString,
     age TypeAge}
TypeAge ::= INTEGER
```

"X", "X1", "X2" et "X3" sont toutes des définitions de types identiques. Les différences d'espaces et de commentaires ne sont pas visibles et l'utilisation de la référence de type "TypeAge" dans "X3" n'affecte pas la définition de ce type. Noter toutefois que si l'un des identificateurs des éléments du type "SEQUENCE" était modifié, les types cesseraient d'être des définitions identiques et il n'y aurait plus de correspondance entre leurs valeurs.

F.7.3 Exemple 2

```

B ::= SET
   {nom VisibleString,
    age INTEGER}
B1 ::= SET
     {age INTEGER,
      nom VisibleString}

```

sont des définitions de types identiques à condition que ni l'une ni l'autre ne soit dans un module dont l'en-tête contient "AUTOMATIC TAGS", sinon il ne s'agit plus de types identiques, et il n'y a plus de correspondance entre leurs valeurs. On peut donner des exemples similaires avec "CHOICE" et "ENUMERATED" (au moyen de la forme "identifier" de "EnumerationItem").

F.7.4 Exemple 3

```

C ::= SET
   {nom [0]VisibleString,
    age INTEGER}
C1 ::= SET
     {nom VisibleString,
      age INTEGER (1..64)}

```

ne sont pas des définitions de types identiques, et ne sont ni l'une ni l'autre des définitions de types identiques à celles de "B" ou "B1", et il n'y a pas de correspondance entre les valeurs de "C" et "C1" ni entre leurs valeurs et celles de "B" ou "B1".

F.7.5 Exemple 4

```

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

```

est légal, et attribue la valeur "3" à "z" au moyen de la correspondance entre valeurs définie au paragraphe F.4.5.

F.7.6 Exemple 5

```

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

```

est légal, et attribue la valeur "{version1, version3}" à "b2".

F.7.7 Exemple 6

D'après les définitions du paragraphe F.1.1, l'utilisation dans des types "SEQUENCE" de notations:

```

X DEFAULT Y

```

est légale, lorsque "X" désigne "A", "B", "C", "D", "E" ou "F" ou le texte figurant dans la partie droite des définitions ainsi nommées, et y désigne "a", "b", "c", "d", "e" ou "f", sauf dans les cas suivants: "E DEFAULT y" est illégal pour "a", "b", "c", "d", "f" et "C DEFAULT e" est illégal car, en pareils cas, il n'existe pas de correspondance entre la référence de valeur par défaut et l'une des valeurs du type par défaut.

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux pour données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information
Série Z	Langages et aspects informatiques généraux des systèmes de télécommunication