

Remplacée par une version plus récente



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

**Amendement 1
X.680**

(04/95)

**RÉSEAUX DE COMMUNICATION DE DONNÉES
ET COMMUNICATION ENTRE SYSTÈMES OUVERTS**

**RÉSEAUTAGE OSI ET ASPECTS
DES 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 1: RÈGLES D'EXTENSIBILITÉ

**Amendement 1 à la
Recommandation UIT-T X.680**
Superseded by a more recent version

(Antérieurement «Recommandation du CCITT»)

Remplacée par une version plus récente

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. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT) (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie 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. Le texte de la Recommandation X.680, Amendement 1, de l'UIT-T a été approuvé le 10 avril 1995. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 8824-1.

NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

© UIT 1996

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.

Remplacée par une version plus récente

RECOMMANDATIONS UIT-T DE LA SÉRIE X

RÉSEAUX DE COMMUNICATION DE DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

(Février 1994)

ORGANISATION DES RECOMMANDATIONS DE LA SÉRIE X

Domaine	Recommandations
RÉSEAUX PUBLICS POUR DONNÉES	
Services et services complémentaires	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
Définition 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 de couche	X.280-X.289
Test de conformité	X.290-X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Considérations générales	X.300-X.349
Systèmes mobiles de transmission de données	X.350-X.369
Gestion	X.370-X.399
SYSTÈMES DE MESSAGERIE	X.400-X.499
ANNUAIRE	X.500-X.599
RÉSEAUTAGE OSI ET ASPECTS DES SYSTÈMES	
Réseautage	X.600-X.649
Dénomination, adressage et enregistrement	X.650-X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680-X.699
GESTION OSI	X.700-X.799
SÉCURITÉ	X.800-X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850-X.859
Traitement des transactions	X.860-X.879
Opérations distantes	X.880-X.899
TRAITEMENT OUVERT RÉPARTI	X.900-X.999

Remplacée par une version plus récente

TABLE DES MATIÈRES

Page

1	Domaine d'application.....	1
2	Références normatives	1
	2.1 Recommandations Normes internationales identiques.....	1
3	Modifications apportées à l'introduction	1
4	Modifications apportées aux définitions	2
5	Modèle ASN.1 de l'extension de type	2
6	Prescriptions d'extensibilité sur les règles de codage	3
7	Modifications apportées à la définition de module	3
8	Modifications apportées à la production d'ensemble d'éléments ValueSet	4
9	Modifications apportées à la définition des types et valeurs	4
10	Modifications apportées à la notation du type énuméré (ENUMERATED).....	4
11	Modifications apportées à la notation des types séquence	5
12	Modifications apportées à la notation des types ensemble (SET)	5
13	Modifications apportées à la notation des types choix (CHOICE)	6
14	Modifications apportées à la notation des types contraints	6
15	Modifications apportées à la notation des identificateurs d'exception	6
16	Modifications apportées à la spécification des ensembles d'éléments.....	7
	Annexe A – Annexe didactique sur le modèle ASN.1 d'extension de type.....	8

Remplacée par une version plus récente

Résumé

Les règles d'extensibilité ASN.1 précisent comment écrire un module ASN.1 pour permettre une évolution progressive vers une nouvelle version des spécifications ASN.1.

Remplacée par une version plus récente

Introduction

La présente Recommandation | Norme internationale explicite les modifications apportées à la Rec. UIT-T X.680 | ISO/CEI 8824-1 pour prendre en charge les règles d'extensibilité ASN.1.

Les règles d'extensibilité ASN.1 précisent comment écrire un module ASN.1 de façon à permettre une évolution progressive vers une nouvelle version de spécification ASN.1. La nouvelle version peut différer de l'ancienne par l'ajout de nouvelles composantes à une production de type ensemble, séquence ou choix, par l'ajout de nouveaux éléments à un type énuméré, par la relaxation de contraintes imposées à la spécification d'un sous-type. Une migration par étapes vers une nouvelle version d'une spécification ASN.1 devra permettre à des entités homologues en communication à travers le réseau d'avoir une connaissance différente de l'ensemble des valeurs permises par la syntaxe abstraite, et d'être quand même capables de communiquer entre elles sans qu'aucune d'elles ne sache si l'entité homologue a une connaissance différente de l'ensemble des valeurs considérées par la spécification de syntaxe abstraite.

Des entités A, B, C et D initialement homologues pourront par exemple avoir des vues identiques des types de valeurs pouvant être échangés. En supposant que la spécification ASN.1 qui décrit ces valeurs a été définie à l'origine en gardant à l'esprit la possibilité d'extension de cette spécification, il sera possible d'effectuer une telle extension en ajoutant par exemple de nouvelles composantes à une production du type séquence, créant ainsi une nouvelle version de spécification ASN.1. Les entités homologues A et B adopteront par exemple immédiatement la nouvelle version, alors que les entités C et D continueront à utiliser l'ancienne, les quatre entités continuant à communiquer entre elles sans difficulté du fait que la production du type séquence a été déclarée à l'origine comme étant extensible. Lorsque les deux entités A et B échangent une valeur du type séquence étendu, elles verront toutes deux les nouvelles composantes ajoutées au type séquence (si celles-ci sont transmises). Si l'entité A (ou B) transmet un message contenant la nouvelle composante à l'entité C (ou D), le message apparaîtra à C au niveau abstrait (c'est-à-dire après décodage) sans les composantes nouvelles; seules les composantes de la structure séquence d'origine seront vues. Si l'entité C (ou D) transmet un message à l'entité A (ou B), le message ne contiendra (bien sûr) aucune des nouvelles composantes; un tel message apparaîtra à l'entité A comme si une entité homologue disposant de la même version de la spécification ASN.1 (l'entité B par exemple) lui avait transmis un message sans aucune des composantes nouvelles. Les entités C et D continueront quant à elles à échanger les messages de la même façon qu'auparavant.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

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

AMENDEMENT 1 (à la Rec. X.680 | ISO/CEI 8824-1)

Règles d'extensibilité

1 Domaine d'application

La présente Recommandation | Norme internationale précise les modifications à apporter à la Rec. UIT-T X.680 | ISO/CEI 8824-1 pour prendre en charge les règles d'extensibilité ASN.1.

2 Références normatives

Les Recommandations et les Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite dans le présent document, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toute Recommandation et Norme internationale étant sujettes à révision, les utilisateurs de la présente Recommandation | Norme internationale sont invités à étudier la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT tient à jour une liste des Recommandations UIT-T en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: Spécification de la notation de base.*
- Recommandation UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: Spécification des contraintes.*
- Recommandation UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: Paramétrage des spécifications de la notation de syntaxe abstraite numéro un.*
- Recommandation UIT-T X.690 (1994) | ISO/CEI 8825-1:1995, *Technologies de l'information – Règles de codage de l'ASN.1: Spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*
- Recommandation UIT-T X.691 (1995) | ISO/CEI 8825-2:1995, *Technologies de l'information – Règles de codage de l'ASN.1: Règles de codage en paquets.*

3 Modifications apportées à l'introduction

{Ajouter le texte suivant à l'introduction de la Rec. UIT-T X.680 | ISO/CEI 8824-1 juste avant le paragraphe commençant par «Les articles 8 à 31 inclus ...»:}

Une spécification ASN.1 sera établie initialement avec un ensemble de types ASN.1 complètement définis. Ultérieurement, il pourra s'avérer nécessaire de modifier ces types (généralement en ajoutant une nouvelle composante à un type séquence ou ensemble). Si une telle modification doit conserver la possibilité pour les applications utilisant les

anciennes définitions de types d'interfonctionner de manière parfaitement définie avec les applications utilisant les nouvelles définitions de types, les règles de codage doivent en fournir les moyens. La notation ASN.1 prend en charge l'inclusion d'un **marqueur d'extension** sur un certain nombre de types. Un tel marqueur signale aux règles de codage l'intention du concepteur de considérer ce type comme un élément appartenant à une suite de types liés (les versions du même type initial) appelée **suite d'extension**, et que les règles d'extension doivent permettre le transfert d'informations entre applications utilisant des types différents mais faisant partie de la même suite d'extension.

4 Modifications apportées aux définitions

{Ajouter les nouvelles définitions suivantes à celles de la Rec. UIT-T X.680 | ISO/CEI 8824-1. Le caractère alphabétique apparaissant après le numéro d'article sera remplacé par le numéro approprié une fois les définitions insérées dans le document de base.}

3.8.a suite d'extension: Suite de types ASN.1 qui peuvent être ordonnés de telle manière que chaque type de la suite soit formé par l'ajout d'un texte à la fin de la notation du type qui le précède immédiatement dans la suite.

NOTE – Il est possible d'étendre les types aussi bien imbriqués que non imbriqués.

3.8.b marqueur d'extension: Indicateur syntaxique (points de suspension) inclus dans tous les types faisant partie d'une suite d'extension.

3.8.c racine d'extension: Type extensible qui est le premier d'une suite d'extension. Il comporte le marqueur d'extension sans autre notation supplémentaire que des commentaires ou des espacements entre le marqueur et la parenthèse «)» ou l'accolade «}» de fermeture de la déclaration de type.

NOTE – Seule une racine d'extension peut être le premier élément d'une suite d'extension.

3.8.d addition d'extension: Une des notations ajoutées dans une série d'extension. Pour les types ensemble et les types séquence, chaque addition d'extension est l'addition d'un élément simple. Pour un type énuméré, il s'agit de l'addition d'un seul élément d'énumération. Pour le type choix, il s'agit de l'addition d'un seul élément de choix supplémentaire. Pour une contrainte, il s'agit de l'addition d'un élément de sous-type.

NOTE – Les additions d'extension sont ordonnées à la fois textuellement (à la suite du marqueur d'extension) et logiquement (leurs valeurs d'étiquetage vont croissant).

3.8.e de même extension: Deux types ayant la même racine d'extension et dont l'un a été créé en ajoutant zéro, une ou plusieurs additions d'extension à l'autre.

3.8.f type extensible: Type comportant un marqueur d'extension.

3.8.g contrainte extensible: Contrainte de sous-type comportant un marqueur d'extension.

5 Modèle ASN.1 de l'extension de type

{Ajouter le texte suivant comme nouvel article avant l'article 6 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, en utilisant l'en-tête du présent article.}

Lors du décodage d'un type extensible, un décodeur peut détecter:

- a) l'absence d'additions d'extension prévues dans le type séquence ou ensemble;
- b) la présence d'additions d'extension imprévues quelconques venant éventuellement s'ajouter au type séquence ou ensemble, ou une alternative inconnue dans un type choix, ou un élément inconnu dans un type énuméré, ou une longueur ou valeur imprévue dans un type de contrainte extensible.

Formellement, une syntaxe abstraite définie par le type extensible «X» ne contient pas seulement les valeurs du type «X», mais aussi les valeurs de tous les types de même extension que «X». Le processus de décodage ne signale donc jamais d'erreur lorsqu'il détecte une des situations ci-dessus a) ou b). Les mesures à prendre dans chacune de ces situations doivent être spécifiées par le concepteur au niveau de la couche d'application.

NOTE – La mesure prise sera fréquemment d'ignorer la présence d'extensions imprévues, et d'utiliser une valeur par défaut ou un indicateur «absent» pour les extensions prévues manquantes.

Les additions d'extension imprévues détectées par le décodeur dans un type extensible peuvent être incluses ultérieurement dans le codage de ce type (pour le renvoyer à l'expéditeur ou le retransmettre à une tierce partie), sous réserve que la même syntaxe de transfert soit utilisée dans la transmission suivante.

6 Prescriptions d'extensibilité sur les règles de codage

{Ajouter le texte suivant comme nouvel article de la Rec. UIT-T X.680 | ISO/CEI 8824-1 avant l'article 6 et après l'article nouvellement créé ci-dessus, en utilisant l'en-tête du présent article.}

6.1 Toutes les règles de codage ASN.1 permettront le codage des valeurs d'un type extensible «X» de telle façon qu'elles puissent être décodées en utilisant un type extensible «Y» de même extension que le type «X». Les règles de codage permettront en outre de recoder par le type «Y» les valeurs de type «X» précédemment décodées avec le type «Y», puis de les décodées avec un troisième type extensible «Z» de même extension que «Y» (et donc que «X»).

NOTE – Les types «X», «Y» et «Z» peuvent apparaître dans un ordre quelconque dans la suite d'extension.

Si une valeur d'un type extensible «X» est codée puis relayée (directement ou par l'intermédiaire d'une application relais utilisant un type «Z» de même extension) vers une autre application qui la décode en utilisant le type extensible «Y» de même extension que «X», le décodeur utilisant le type «Y» obtiendra une valeur abstraite composée des éléments suivants:

- a) une valeur abstraite du type racine d'extension;
- b) une valeur abstraite de chaque addition d'extension présente à la fois dans les types «X» et «Y»;
- c) un codage déterminé de chaque addition d'extension (s'il y en a) présente dans «X» mais pas dans «Y».

Les codages de l'alinéa c) pourront être inclus dans un codage ultérieur de la valeur de type «Y» si l'application le nécessite. Un tel codage sera un codage valide d'une valeur du type «X».

Exemple: supposons qu'un système A utilise un type racine extensible (le type «X») de type séquence ou ensemble avec une extension de type entier optionnel, alors que le système B utilise un type de même extension (le type «Y») ayant deux extensions de type entier optionnel. Si le système B transmet alors une valeur du type «Y» omettant la valeur entière de la première extension et comprenant celle de la seconde, le système A ne doit pas confondre le codage reçu avec celui d'une valeur de type «X» comprenant la première (et unique) addition d'extension. De plus, A doit être capable de recoder la valeur de «X» avec une valeur présente pour le premier entier, suivi de la deuxième valeur entière reçue de B si le protocole d'application le nécessite.

6.2 Toutes les règles de codage ASN.1 spécifieront le codage et le décodage d'une valeur de type énuméré ou de type choix de telle manière que si la valeur transmise appartient à l'ensemble des additions d'extension possédées en commun par le codeur et le décodeur, elle puisse être décodée avec succès, et si elle n'appartient pas à cet ensemble, que le décodeur puisse en délimiter le codage et l'identifier comme une valeur appartenant à une addition d'extension (inconnue).

6.3 Toutes les règles de codage ASN.1 spécifieront le codage et le décodage de types soumis à des contraintes extensibles de telle manière que si la valeur transmise appartient à l'ensemble des additions d'extension possédées en commun par le codeur et le décodeur, elle puisse être décodée avec succès, et si elle n'appartient pas à cet ensemble, que le décodeur puisse en délimiter le codage et l'identifier comme une valeur appartenant à une addition d'extension (inconnue).

Dans tous les cas, la présence d'additions d'extension n'affectera en rien la possibilité de reconnaître les derniers éléments lorsqu'un type muni d'un marqueur d'extension est imbriqué dans un autre type quelconque.

NOTE – Toutes les variantes des règles de codage de base BER et des règles de codage compactes PER de l'ASN.1 satisfont à ces prescriptions.

7 Modifications apportées à la définition de module

{Modifier les productions du paragraphe 10.1 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 de la manière suivante:}

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

```
BEGIN
ModuleBody
END
```

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

{Les autres productions du paragraphe 10.1 restent inchangées.}

{Insérer un nouveau paragraphe après le paragraphe 10.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

10.3 bis L'option «EXTENSIBILITY IMPLIED» est équivalente à l'insertion d'un marqueur d'extension (...) dans le texte de la définition de chaque type, dans un module pour lequel cette option est permise. Si «EXTENSIBILITY IMPLIED» ne figure pas, l'extensibilité n'est assurée que pour les types du module où figure explicitement un marqueur d'extension.

NOTE – L'option «EXTENSIBILITY IMPLIED» ne s'applique qu'à des types. Elle n'a pas d'incidence sur les ensembles d'objets.

8 Modifications apportées à la production d'ensemble d'éléments ValueSet

{Remplacer la production ValueSet du paragraphe 13.5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 par:}

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

9 Modifications apportées à la définition des types et valeurs

{Insérer le (nouveau) paragraphe 14.12 dans la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

14.12 La présence implicite ou explicite d'un marqueur d'extension dans la définition d'un type n'a aucun effet sur la notation d'une valeur de ce type. En d'autres termes, la notation de la valeur d'un type comportant un marqueur d'extension est exactement la même que si le marqueur était absent.

10 Modifications apportées à la notation du type énuméré (ENUMERATED)

{Remplacer la production de l'article 17 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 par:}

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

RootEnumeration ::= Enumeration

AdditionalEnumeration ::= Enumeration
```

{La définition du type «Enumeration» reste inchangée.}

{Modifier la Note 2 du paragraphe 17.1, pour avoir:}

2 Les valeurs numériques incluses dans les nombres nommés «NamedNumber» d'une séquence «RootEnumeration» ne sont pas nécessairement ordonnées ni consécutives; les valeurs numériques incluses dans les nombres nommés «NamedNumber» d'une séquence «AdditionalEnumeration» ne sont pas nécessairement consécutives.

{Ajouter le texte suivant après le paragraphe 17.3:}

17.3 bis Les items «AdditionalEnumeration» ajoutés successivement dans un type doivent avoir des valeurs croissantes.

17.3 ter Lorsqu'un «NamedNumber» est utilisé dans la définition d'une «AdditionalEnumeration», sa valeur doit être différente de celles de tous les items «EnumerationNumber» précédemment définis dans ce type, que ces derniers se trouvent dans la racine d'énumération ou non. Par exemple:

```
A ::= ENUMERATED {a, b, ..., c(0)}           -- non valide, puisque 'a' et 'c' = 0
B ::= ENUMERATED {a, b, ..., c, d(2)}       -- non valide, puisque 'c' et 'd' = 2
C ::= ENUMERATED {a, b(3), ..., c(1)}       -- valide, 'c' = 1
D ::= ENUMERATED {a, b, ..., c(2)}         -- valide, 'c' = 2
```

17.3 quater La valeur associée au premier énuméré supplémentaire «AdditionalEnumeration» de type identificateur (et non pas «NamedNumber») doit être la plus petite valeur (a) pour laquelle aucun «EnumerationItem» n'est défini dans la «RootEnumeration» et (b) à laquelle sont inférieurs, le cas échéant, tous les «EnumerationItem» de la «AdditionalEnumeration». Par exemple, les trois productions suivantes sont valides:

```
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 Modifications apportées à la notation des types séquence

{Remplacer la production de l'article 22 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 par:}

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

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

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

RootComponentTypeList ::= ComponentTypeList

AdditionalComponentTypeList ::= ComponentTypeList
```

{La définition du type «ComponentTypeList» reste inchangée.}

NOTE – Les types composantes «ComponentType» qui ne sont pas déclarés OPTIONAL ou DEFAULT dans la liste «AdditionalComponentTypeList» doivent toujours être codés, sauf lorsque la valeur de données de présentation est relayée depuis un expéditeur utilisant une version plus ancienne de la syntaxe abstraite, auquel cas le type «ComponentType» n'est pas défini.

{Remplacer la *première phrase* du paragraphe 22.4 par ce qui suit:}

Le type indiqué dans la notation «COMPONENTS OF Type» sera un type séquence ne contenant pas lui-même de marqueur d'extension, mais qui pourra comporter des composantes avec un tel marqueur.

{Insérer le nouveau paragraphe suivant après le paragraphe 22.4:}

22.4 bis La notation «COMPONENTS OF Type» ne sera pas utilisée dans une liste «AdditionalComponentTypeList».

12 Modifications apportées à la notation des types ensemble (SET)

{Remplacer la production de l'article 24 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 par:}

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

La notation «ComponentTypeLists» est spécifiée au paragraphe 22.1.

{Remplacer la *première phrase* du paragraphe 24.2 par ce qui suit:}

Si le type figurant dans une notation «COMPONENTS OF Type» comporte un marqueur d'extension, celui-ci ne sera pas visible depuis la notation de composante. Aux fins de la notation «COMPONENTS OF Type», le type y figurant sera donc traité comme s'il ne comportait pas de marqueur d'extension.

{Ajouter le texte suivant après le paragraphe 24.3:}

24.3 bis Les étiquettes des «ComponentType» successifs de la «AdditionalComponentTypeList» doivent avoir des valeurs croissantes, selon l'ordre canonique (voir 6.4) et doivent se présenter successivement en dernier dans le type ensemble.

{Modifier le paragraphe 24.4, pour avoir:}

24.4 Les paragraphes 22.2, 22.2 *bis* ainsi que 22.6 à 22.10, s'appliquent également aux types ensemble.

13 Modifications apportées à la notation des types choix (CHOICE)

{Remplacer la production de l'article 26 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 par:}

```
ChoiceType ::= CHOICE "{" AlternativeTypeLists "}"
AlternativeTypeLists ::=
    RootAlternativeTypeList |
    RootAlternativeTypeList "," ExtensionAndException |
    RootAlternativeTypeList "," ExtensionAndException "," AdditionalAlternativeTypeList

RootAlternativeTypeList ::= AlternativeTypeList

AdditionalAlternativeTypeList ::= AlternativeTypeList
```

{La définition du type «AlternativeTypeList» reste inchangée.}

{Insérer le paragraphe suivant après le paragraphe 26.3:}

26.3 bis Chaque type nommé «NamedType» ajouté à la liste «AdditionalAlternativeTypeList» recevra une étiquette dont le numéro sera canoniquement supérieur à ceux des autres composantes de la liste, et se placera après tous les autres types nommés dans le type choix.

{Dans le paragraphe 26.5, remplacer «Lorsque le type choix est utilisé ...» par:}

Lorsque le type choix ne comporte pas de marqueur d'extension et qu'il est utilisé ...

{Ajouter la phrase suivante à la suite de la première phrase du paragraphe 26.5:}

Lorsque le type choix comporte un marqueur d'extension, il ne sera pas utilisé là où la présente Norme internationale impose des étiquettes distinctes (voir 22.5, 24.3 et 26.2).

14 Modifications apportées à la notation des types contraintes

{Remplacer les deux dernières phrases du paragraphe 42.5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 par:}

La spécification d'exception «ExceptionSpec» est définie à l'article 43. A moins qu'elle ne soit utilisée avec un «marqueur d'extension» (voir l'article 44 *bis*), elle n'apparaîtra que si la spécification de contrainte «ConstraintSpec» comporte une occurrence de référence muette «DummyReference» (voir 8.4 de la Rec. UIT-T X.683 | ISO/CEI 8824-4) ou s'il s'agit d'une contrainte définie par l'utilisateur «UserDefinedConstraint» (voir l'article 9 de la Rec. UIT-T X.682 | ISO/CEI 8824-3).

{Remplacer la production du paragraphe 42.6 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 par:}

```
SubtypeConstraint ::= ElementSetSpecs
```

15 Modifications apportées à la notation des identificateurs d'exception

{Ajouter les nouveaux paragraphes 43.5 et 43.6 à la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

43.5 Lorsqu'un type est soumis à une contrainte multiple et que plusieurs de ces contraintes possèdent un identificateur d'exception, l'identificateur associé à la contrainte la plus extérieure sera considéré comme l'identificateur d'exception de ce type.

43.6 Lorsqu'un marqueur d'exception apparaît dans un type intervenant dans des opérations ensemblistes, l'identificateur d'exception est ignoré et n'est pas hérité par le type contraint résultant de ces opérations.

16 Modifications apportées à la spécification des ensembles d'éléments

{Ajouter les éléments suivants juste avant la production «ElementSetSpec» du paragraphe 44.1 de la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

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

```
RootElementSetSpec ::= ElementSetSpec
```

```
AdditionalElementSetSpec ::= ElementSetSpec
```

NOTE – Les éléments représentés par l'ensemble «ElementSetSpecs» sont la réunion des éléments représentés par «RootElementSetSpec» et par «AdditionalElementSetSpec».

{Ajouter le nouveau paragraphe 44 bis et les paragraphes suivants à la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

44 bis Marqueur d'extension

NOTE – Comme toute notation de contrainte en général, le marqueur d'extension n'a pas d'effet sur certaines règles de codage de l'ASN.1, telles que les règles de codage de base, mais en affecte d'autres telles que les règles de codage condensé.

44.1 Le marqueur d'extension (les points de suspension) indique que des additions d'extension sont prévues. Il n'indique en rien la manière de traiter de telles additions, sauf qu'elles ne doivent pas être traitées comme des erreurs pendant le processus de décodage.

44.2 L'utilisation conjointe du marqueur d'extension et d'un identificateur d'exception indique que des additions d'extension sont prévues, qu'elles ne doivent pas être traitées comme une erreur pendant le processus de décodage, et que les normes d'application prescrivent de prendre des mesures particulières au niveau de l'application si une contrainte est violée. Il est recommandé d'utiliser cette notation dans les situations où intervient un fonctionnement en enregistrement et retransmission ou toute autre forme de relayage pour demander de retourner toute addition d'extension non reconnue à l'application expéditrice pour qu'elle procède éventuellement à son recodage et à son renvoi.

44.3 Les opérations ensemblistes seront exécutées s'il y a lieu sur les ensembles «ElementSetSpecs» sans tenir compte de la présence de marqueurs d'extension.

NOTE – En d'autres termes, la présence d'un marqueur d'extension n'affecte en rien le résultat d'une opération ensembliste.

44.4 S'il est fait référence, dans un «ContainedSubtype», à un type défini avec une contrainte extensible, le nouveau type ainsi défini n'hérite pas du marqueur d'extension. Pour être considéré comme extensible, le nouveau type doit comporter un marqueur d'extension explicite dans ses «ElementSetSpecs». Par exemple:

```
A ::= INTEGER (0..10, ...)      -- A est extensible
B ::= INTEGER (A)              -- B n'est pas extensible
C ::= INTEGER (A, ...)         -- C est extensible
```

44.5 Si une contrainte comportant des «ElementSetSpecs» ne contenant pas de marqueur d'extension est appliquée à un type défini avec une contrainte extensible, le type résultant se comporte comme s'il était doté d'une contrainte non extensible. Par exemple:

```
A ::= INTEGER (0..10, ...)      -- A est extensible
B ::= A (2..5)                 -- B n'est pas extensible
C ::= A                        -- C est extensible
```

44.6 Les composants d'un type ensemble, séquence ou choix, dont la contrainte est de ne pas figurer, ne doivent pas figurer, que ledit type ensemble, séquence ou choix soit extensible ou non.

NOTE – Les contraintes sur les types imbriqués intérieurs n'affectent pas l'extensibilité.

Exemple:

```
A ::= SEQUENCE {
  a  INTEGER,
  b  BOOLEAN OPTIONAL,
  ...
}
B ::= A (WITH COMPONENTS {b ABSENT})  -- B est extensible, mais 'b' ne figurera
                                       -- dans aucune de ses valeurs
```

{Ajouter l'annexe didactique suivante à la Rec. UIT-T X.680 | ISO/CEI 8824-1:}

Annexe A

Annexe didactique sur le modèle ASN.1 d'extension de type

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

A.1 Aperçu général

A.1.1 Il peut arriver qu'un type ASN.1 évolue au cours du temps à partir d'une **racine d'extension** en se voyant adjoindre une suite d'extensions appelées **additions d'extension**.

A.1.2 Le type ASN.1 connu d'une réalisation particulière peut être le type racine d'extension, ou aussi le type racine d'extension enrichi d'une ou plusieurs additions d'extension. Chacun des types ASN.1 contenant une telle addition d'extension contient également toutes les additions d'extension précédemment définies.

A.1.3 Les types ASN.1 définis d'une telle suite sont dits **de même extension** (voir le paragraphe 3.8.e pour une définition plus précise de ce qualificatif), et il est prescrit que les règles de codage codent les types de même extension de telle manière que si deux systèmes en communication utilisent deux types différents mais de même extension, ils puissent se transmettre avec succès les contenus informationnels correspondant à la partie de type qu'ils ont en commun. Il est également prescrit que les parties de type qui ne sont pas communes aux deux systèmes puissent être délimitées et retransmises (peut-être à une partie tierce) au cours d'une transmission ultérieure, sous réserve que la même syntaxe de transfert soit utilisée.

NOTE – L'expéditeur peut utiliser un type antérieur ou postérieur au type utilisé par le destinataire dans la suite d'extension.

A.1.4 La suite de types obtenue par enrichissement progressif d'un type racine s'appelle une **suite d'extension**. Pour que les règles de codage soient à même de transmettre des types de même extension (ce qui peut nécessiter la transmission d'un surcroît de données sur la ligne), de tels types (y compris le type racine) doivent être syntaxiquement marqués. Le marqueur adopté, appelé **marqueur d'extension**, est constitué de trois points de suspension (...).

EXEMPLE:

Type racine	1 ^{re} extension	2 ^e extension
<pre>SEQUENCE { a INTEGER, ... }</pre>	<pre>SEQUENCE { a INTEGER, ..., b BOOLEAN, c INTEGER }</pre>	<pre>SEQUENCE { a INTEGER, ..., b BOOLEAN, c INTEGER, d REAL }</pre>

A.1.5 La spécification du type racine d'extension apparaît avant et au-dessus du marqueur d'extension ASN.1, et les additions d'extension apparaissent après et en dessous de celui-ci.

A.1.6 Un type possédant un marqueur d'extension peut être imbriqué dans un type qui n'en possède pas, ou dans la racine d'extension d'un type qui en possède, ou encore dans une addition d'extension d'un tel type. Dans ce cas, la suite d'extension est traitée indépendamment, et le type imbriqué muni de son marqueur d'extension n'a aucun impact sur le type dans lequel il est imbriqué. Un seul marqueur d'extension (les points de suspension) peut apparaître dans une structure donnée.

A.1.7 Un nouveau type dans la suite d'extension est défini en ajoutant une seule addition d'extension au type précédent dans la suite. Ceci n'exclut pas l'ajout de multiples additions d'extension dans le cadre de publications successives d'une Recommandation | Norme internationale donnée.

A.1.8 Si en pratique les additions d'extension viendront s'ajouter au fil du temps, la spécification et le modèle ASN.1 sous-jacents ne font pas intervenir le temps. Deux types sont dits de même extension si l'un peut être dérivé de l'autre par des additions d'extension. En d'autres termes, l'un doit contenir toutes les composantes de l'autre. On peut imaginer des types qui décroissent au fil du temps, même si un tel cas de figure est peu vraisemblable. On peut même imaginer qu'un type *commence* avec une quantité d'additions d'extension qui sont retirées au fil du temps! Mais la seule chose prise en considération par l'ASN.1 et ses règles de codage est de savoir si un couple de types spécifiés sont oui ou non de même extension. S'ils le sont, les règles de codage de l'ASN.1 assureront **dans leur ensemble** l'interfonctionnement des entités utilisatrices.

A.1.9 Prenons un type et décidons si nous voulons que l'interfonctionnement futur avec les premières versions soit assuré au cas où une extension de ce type s'imposerait ultérieurement. Si tel est le cas, il faut inclure **d'emblée** un marqueur d'extension. Il devient alors possible d'ajouter ultérieurement des additions d'extension au type sans modifier le train binaire transmis pour les valeurs antérieures, tout en assurant un traitement défini des valeurs étendues par les systèmes antérieurs. Il est toutefois important de noter que l'ajout d'un marqueur d'extension à un type qui en était dépourvu (ou la suppression d'un tel marqueur) **modifiera** généralement le train binaire transmis et interdira l'interfonctionnement des systèmes antérieurs et postérieurs à une telle modification. Une telle modification imposera généralement le changement du numéro de version de tous les protocoles affectés.

A.1.10 Le tableau suivant indique les types ASN.1 qui peuvent former le type racine d'une suite d'extension, et la nature de l'addition d'extension simple autorisée pour ce type (des additions d'extension multiples peuvent bien sûr être opérées successivement ou en même temps).

Additions d'extension

Type racine d'extension	Nature des additions d'extension
Type énuméré (ENUMERATED)	Addition d'un élément énuméré supplémentaire simple à la fin des énumérations additionnelles «AdditionalEnumeration», avec une valeur d'énumération supérieure à toute énumération déjà ajoutée.
Type séquence (SEQUENCE) et ensemble (SET)	Addition d'une composante «Component» simple à la fin de la liste «AdditionalComponentTypeList». A noter qu'il n'existe aucune prescription imposant d'accompagner le type composant «ComponentType» d'une déclaration «OPTIONAL» ou «DEFAULT», même si ce sera souvent le cas.
Type choix (CHOICE)	Addition d'un type nommé «NamedType» unique, à la fin de la liste «AdditionalAlternativeTypeList».
Notation des contraintes	Addition d'une spécification «AdditionalElementSetSpec» unique, à la notation de l'ensemble de spécifications «ElementSetSpecs».

A.2 Effets sur le numérotage des versions, etc.

A.2.1 Lorsqu'une spécification ASN.1 est republiée avec des définitions de types et des spécifications d'ensembles d'objets modifiées par extension, de telles modifications ne nécessiteront pas en elles-mêmes le changement de l'identificateur d'objet du module ou du numéro de version du protocole.

A.2.2 De telles modifications pourront être accompagnées pour d'autres raisons par le changement du numéro de version, mais il n'y a là aucune obligation.

A.2.3 A l'inverse, l'ajout d'un marqueur d'extension à un type qui en était précédemment dépourvu, ou l'ajout de composantes à un type séquence ou ensemble (par exemple) sans marqueur d'extension, génère un nouveau type qui n'est **pas** de même extension que l'ancien type; le module le contenant doit alors recevoir un nouvel identificateur d'objet, et le protocole associé un nouveau numéro de version.

A.3 Prescriptions imposées aux règles de codage

A.3.1 Une syntaxe abstraite peut être définie comme les valeurs d'un même type ASN.1 extensible. Elle contient alors toutes les valeurs pouvant être obtenues par l'addition ou la suppression d'additions d'extension. Une telle syntaxe abstraite est appelée syntaxe abstraite de coextension.

A.3.2 Un ensemble homogène de règles de codage pour une syntaxe abstraite de coextension satisfait aux prescriptions supplémentaires suivantes.

NOTE – Toutes les règles de codage ASN.1 satisfont à ces prescriptions.

A.3.3 La définition des procédures de transformation d'une valeur abstraite en une forme codée aux fins de transfert, et des procédures de transformation d'une forme codée reçue en une valeur abstraite, tiendra compte de la possibilité que l'expéditeur et le destinataire utilisent des syntaxes abstraites différentes mais de même extension.

A.3.4 Dans ce cas, les règles de codage garantiront que lorsque l'expéditeur dispose d'une spécification de type antérieur dans la suite d'extension à celle du destinataire, les valeurs transmises par l'expéditeur seront transférées de telle manière que le destinataire puisse déterminer l'absence d'additions d'extension.

A.3.5 Les règles de codage garantiront que lorsque l'expéditeur dispose d'une spécification de type postérieur dans la suite d'extension à celle du destinataire, il lui sera possible de transmettre des valeurs de ce type au destinataire.

