

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.692

Amendement 1
(08/2004)

SÉRIE X: RÉSEAUX DE DONNÉES, COMMUNICATION
ENTRE SYSTÈMES OUVERTS ET SÉCURITÉ

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

Technologies de l'information – Règles de codage
ASN.1: spécification de la notation de contrôle de
codage (ECN)

**Amendement 1: Prise en charge de
l'extensibilité**

Recommandation UIT-T X.692 (2002) – Amendement 1

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

RÉSEAUX PUBLICS DE 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.369
Réseaux à protocole Internet	X.370–X.379
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.889
Applications génériques de l'ASN.1	X.890–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999
SÉCURITÉ DES TÉLÉCOMMUNICATIONS	X.1000–

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

**Technologies de l'information – Règles de codage ASN.1: spécification de la
notation de contrôle de codage (ECN)**

Amendement 1
Prise en charge de l'extensibilité

Résumé

Le présent amendement inclut la capacité d'utiliser la notation de contrôle de codage (ECN, *encoding control notation*) pour spécifier la manière dont les types ouverts sont codés. Il inclut également la prise en charge de l'amélioration du mécanisme des "conditions", qui permet d'exprimer des conditions plus complexes.

Source

L'Amendement 1 de la Recommandation UIT-T X.692 (2002) a été approuvé le 29 août 2004 par la Commission d'études 17 (2001-2004) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8. Un texte identique est publié comme Norme Internationale ISO/CEI 8825-3, Amendement 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.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude 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 1 de l'AMNT.

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, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

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 2005

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

TABLE DES MATIÈRES

	<i>Page</i>
1 Paragraphe 3.2.8	1
2 Nouveaux paragraphes 9.25 <i>bis</i> et 9.25 <i>ter</i>	1
9.25 <i>bis</i> Autres conditions d'application des codages.....	1
9.25 <i>ter</i> Contrôle de codage pour le type ouvert.....	1
3 Paragraphe 13.2.9.....	2
4 Paragraphe 13.2.10.5.....	2
5 Paragraphe 17.5.15.....	3
6 Paragraphe 18.2.6.....	3
7 Paragraphe 21.11.1.....	3
8 Paragraphe 21.11.4.....	4
9 Paragraphe 21.11.5.....	4
10 Nouveaux paragraphes 21.11 <i>bis</i>	4
21.11 <i>bis</i> Le type Comparison.....	4
11 Paragraphe 21.12.1.....	5
12 Paragraphe 21.12.4.....	5
13 Nouveau paragraphe 21.12.5.....	5
14 Nouveau paragraphe 21.16.....	5
21.16 Le type IntegerMapping.....	5
15 Paragraphe 23.2.3.8.....	5
16 Paragraphe 23.4.3.8.....	6
17 Paragraphe 23.6.2.3.....	6
18 Paragraphe 23.7.1.....	6
19 Paragraphe 23.7.2.2.....	8
20 Paragraphe 23.7.2.4.....	8
21 Paragraphe 23.7.2.6, 23.7.2.7 et 23.7.2.8.....	8
22 Paragraphe 23.9.3.8.....	8
23 Nouveau paragraphe 23.9 <i>bis</i>	8
23.9 <i>bis</i> Définition des objets de codage pour les classes de la catégorie des types ouverts.....	9
24 Paragraphe 23.12.2.3.....	11
25 Paragraphe 23.13.1.....	11
26 Paragraphes 23.13.2.1 et 23.13.2.2.....	13
27 Paragraphe 23.15.....	13
23.15 Définition des objets de codage pour classes d'autres catégories.....	13
28 Paragraphe 24.3.1.....	14
29 Nouvea paragraphe 24.3.2 <i>bis</i>	14
30 Nouveau paragraphe 24.3.8 <i>bis</i>	14
31 Tableau 6.....	14
32 Paragraphe C.1.....	14
33 Paragraphe C.4.....	15
34 Paragraphe G.2.4.....	15

NORME INTERNATIONALE
RECOMMANDATION UIT-T

Technologies de l'information – Règles de codage ASN.1:
Spécification de la notation de contrôle de codage (ECN)

Amendement 1
Prise en charge de l'extensibilité

NOTE – Tout texte nouveau ou modifié dans le présent amendement est souligné dans les paragraphes à remplacer. En cas d'adjonction de nouveaux paragraphes comportant un titre, seul le titre est souligné. Les parties de texte supprimées sont conservées, mais elles sont biffées d'un trait fin. Lorsque ces parties de texte seront incorporées dans la Recommandation | Norme internationale de base, le soulignement et les passages biffés seront supprimés.

1 **Paragraphe 3.2.8**

Remplacer le § 3.2.8 par le texte suivant:

3.2.8 codage conditionnel: codage qui doit être appliqué seulement si une condition spécifiée ~~aux limites ou une condition de catégorie de longueur~~ est satisfaite.

NOTE – Il peut s'agir d'une condition aux limites, d'une condition de catégorie de longueur ou d'autres conditions plus complexes.

2 **Nouveaux paragraphes 9.25 bis et 9.25 ter**

Ajouter les nouveaux § 9.25bis et 9.25ter et actualiser en conséquence la table des matières:

9.25 bis Autres conditions d'application des codages

9.25 bis.1 Différentes conditions peuvent être testées afin de sélectionner un codage approprié, parmi lesquelles la valeur réelle et la catégorie de limites.

9.25 bis.2 On peut aussi exiger que toutes les conditions répertoriées pour la circonstance soient réalisées.

9.25 bis.3 Un essai portant sur une condition utilise soit une seule valeur d'énumération (telle que "**bounded-without-negatives**") qui contient l'intégralité de l'essai dans la spécification de cette énumération unique, soit une combinaison de trois énumérations.

9.25 bis.4 En cas d'utilisation de trois énumérations, la première identifie l'item qui est soumis à l'essai (par exemple "**test-upper-bound**"), la seconde indique la nature de l'essai (par exemple "**greater-than**") et la troisième spécifie une valeur entière pour l'essai.

9.25 ter Contrôle de codage pour le type ouvert

9.25 ter.1 Les types ouverts constituent souvent un moyen d'extensibilité utilisant un champ d'identification, de nouvelles valeurs pour le champ d'identification et de nouveaux types pour le type ouvert étant ajoutés dans les versions successives (et souvent mis à disposition pour les extensions propres au vendeur).

9.25 ter.2 Ces deux fonctionnalités permettent de demander à un décodeur de décoder un type ouvert lorsque l'implémentation considérée ignore le type qui a été codé dans ce type ouvert.

9.25 ter.3 Le codage pris en charge pour le type ouvert est le même que pour la plupart des autres classes de la catégorie des champs binaires, mais avec la capacité supplémentaire de spécifier qu'un ensemble d'objets de codage différent doit être appliqué au type qui doit être codé dans le type ouvert.

NOTE – Cela est dû au fait que de nombreux protocoles utilisent de préférence un mode de codage différent (souvent fondé sur le principe "type, longueur, valeur") pour le type contenu dans un type ouvert, tout en conservant un mode de codage plus compact pour les champs du message contenant le type ouvert.

9.25 ter.4 Le modèle utilisé pour le codage d'un type ouvert tient compte non seulement du fait qu'un décodeur ne saura pas quel type est contenu dans le type ouvert (les contraintes tabulaires et relationnelles étant invisibles pour les règles de codage compact (PER) ou pour la notation de contrôle de codage (ECN)), mais aussi du fait que l'application sera peut-être à même de déterminer la nature de ce type à partir d'un autre champ figurant dans le protocole, ou dans un message précédent ou (pour les adjonctions propres au vendeur) d'après l'adresse appelante.

9.25 ter.5 Le modèle est donc le suivant: après avoir pris en charge le prébourrage spécifié, et déterminé l'espace de codage ainsi que la valeur éventuelle de prébourrage et de postbourrage, le décodeur demandera à l'application de lui indiquer le type qui a été codé. (Dans le cas d'outils, l'application aura presque certainement préconfiguré l'outil à l'aide d'une liste de types connus pouvant être présents, et n'aura qu'à renvoyer un pointeur indiquant l'un d'entre eux.) Le décodage peut maintenant continuer normalement.

9.25 ter.6 L'application peut, toutefois, indiquer dans sa réponse "type inconnu" (voir le § 9.25 ter.4) et, dans ce cas, le décodeur doit savoir comment déterminer la fin de ce codage inconnu. A cette fin, on active le spécificateur ECN, dans le cas considéré ici, pour disposer d'une structure de codage, et (à titre facultatif) d'un ensemble d'objets de codage à utiliser avec cette structure, que devront utiliser les décodeurs pour décoder les types inconnus dans le type ouvert. La syntaxe à utiliser à cette fin est définie au § 23.

NOTE – Un exemple d'une telle structure de codage pourrait être une structure qui spécifie un codage communément appelé codage "type, longueur, valeur", dont on peut déterminer la fin sans connaître le type qui est en cours de codage.

3 Paragraphe 13.2.9

Remplacer le § 13.2.9 par le suivant:

13.2.9 A des étapes ultérieures de ces procédures, le point d'application peut être un des suivants:

- a) un nom de classe de codage. Ce nom est complètement codable au moyen de la spécification contenue dans un objet de codage de la même classe (voir § 17.1.7);
- b) un constructeur de codage (voir § 16.2.12). Les procédures de construction peuvent être déterminées par la spécification contenue dans un objet de codage de la classe des constructeurs de codage, mais cet objet de codage ne détermine pas le codage des composants. La spécification de l'objet de codage qui est appliquée peut nécessiter qu'un ou plusieurs des composants du constructeur soient remplacés par d'autres structures (paramétrées) avant que le point d'application passe aux composants;
- c) une classe de la catégorie des chaînes de bits ou d'octets qui possède un type contenu en tant que propriété associée aux valeurs (voir § 11.3.4.3 d). Le codage du type contenu ~~dépend de savoir si varie selon qu'une~~ construction **ENCODED BY** est présente et ~~de selon~~ l'application de la spécification de l'objet de codage (voir § 22.11);
- d) une classe de la catégorie des types ouverts. Le codage du composant du type ouvert varie selon qu'une construction **ENCODED WITH** est présente et selon l'application de la spécification de l'objet de codage (voir § 23.9 bis.2);
- ~~e)~~ un composant qui est une classe de codage (éventuellement précédée par une ou plusieurs classes de la catégorie des étiquettes), suivie par une classe de codage dans la catégorie des offres d'options. Les procédures et codages permettant de déterminer la présence ou l'absence sont déterminés par la spécification contenue dans un objet de codage de la classe de la catégorie des offres d'options. Cet objet de codage peut également nécessiter le remplacement de la classe de codage (ainsi que toutes ses classes précédentes dans la catégorie des étiquettes) par une structure (paramétrée) de remplacement avant que cette classe soit codée. Le point d'application passe alors à la première classe de la catégorie des étiquettes (éventuelle), ou au composant, ou à son remplacement;
- ~~ef)~~ une classe de codage précédée par une classe de codage de la catégorie des étiquettes. Les numéros d'étiquette associés à la classe de la catégorie des étiquettes sont codés au moyen de la spécification contenue dans un objet de codage de la classe de la catégorie des étiquettes et le point d'application passe ensuite à la classe étiquetée;
- ~~fg)~~ toute autre classe de codage intégrée. Cet élément est complètement codable au moyen de la spécification contenue dans un objet de codage de cette classe.

4 Paragraphe 13.2.10.5

Remplacer la NOTE figurant dans le § 13.2.10.5 par la suivante:

NOTE – Si l'objet de codage appliqué à une classe de la catégorie des types ouverts contient une construction **ENCODED WITH**, cela détermine l'ensemble d'objets de codage qui est appliqué au composant. Autrement, l'ensemble d'objets de codage combinés appliqué à cette classe est appliqué au composant (voir § 23.9 bis.2).

5 Paragraphe 17.5.15

Remplacer le § 17.5.15 par le suivant:

17.5.15 Si une référence **REFERENCE** est nécessaire en tant que paramètre réel d'un quelconque des objets de codage ou ensembles d'objets de codage utilisés dans cette production, dans ce cas elle peut soit être fournie en tant que paramètre fictif de l'objet de codage qui est défini, ou être fournie en tant que liste "ComponentIdList" (voir le § 15.3.1 pour la syntaxe de la liste "ComponentIdList" – la signification de la liste "ComponentIdList" dans ce contexte est spécifiée ci-dessous).

17.5.15 bis Si le gouverneur n'est pas un constructeur de la catégorie des répétitions, dans ce cas le premier (ou seul) identificateur "identifier" de la liste "ComponentIdList" doit être l'identificateur "Identifier" d'un "NamedType" textuellement présent (à un certain niveau d'emboîtement – voir le § 17.5.15 ter) de la construction qui est obtenue par référencement du gouverneur. Cet identificateur identifie l'entière définition de ce composant "NamedType", que cette définition soit textuellement présente ou non.

17.5.15 ter S'il existe plusieurs identificateurs concordants de ce type, dans ce cas l'identificateur concordant choisi doit être déterminé par la première concordance lors d'une exploration (dans l'ordre du texte) des identificateurs du niveau extérieur, puis par une exploration (dans l'ordre du texte) des identificateurs du deuxième niveau, puis par une exploration (dans l'ordre du texte) des identificateurs du troisième niveau, et ainsi de suite.

17.5.15 quarto Chaque identificateur "identifier" suivant de la liste "ComponentIdList" (le cas échéant) doit être un identificateur "identifier" dans un composant "NamedType" de la structure identifiée par la partie précédente de la liste "ComponentIdList" et identifie l'entière définition de ce composant "NamedType", qu'il soit textuellement présent ou non dans la définition de la structure identifiée par la partie précédente de la liste "ComponentIdList".

17.5.15 quinto Si le gouverneur est un constructeur de la catégorie des répétitions, dans ce cas le paramètre réel pour la référence **REFERENCE** doit être une liste "ComponentIdList" dont le premier identificateur "identifier" identifie un composant qui est textuellement présent dans la structure "EncodingStructure" contenue dans la structure "RepetitionStructure" obtenue par déréférencement de la répétition (voir § 17.5.15 ter). Dans ce cas, les § 17.5.15 ter et 17.5.15 quarto s'appliquent.

17.5.15 sexto Si la référence **REFERENCE** est requise afin de désigner un conteneur, elle peut également être fournie comme suit:

- a) **STRUCTURE** (à condition que le constructeur de la structure codée ne soit pas une catégorie d'options) lorsque cette référence renvoie à cette structure;
- b) **OUTER** lorsque cette référence renvoie au conteneur du codage complet.

NOTE – La production "EncodeStructure" est la seule production dans laquelle des références **REFERENCE** peuvent être fournies, sauf au moyen de paramètres fictifs ou de la construction **OUTER**, ou lorsque les références visent un **flag-to-be-used** ou un **flag-to-be-set** dans la définition d'un objet de codage pour une classe de la catégorie des répétitions qui utilise le remplacement.

6 Paragraphe 18.2.6

Remplacer la Note du § 18.2.6 par la suivante:

NOTE – L'ensemble d'objets de codage combinés appliqués par ces objets de codage au type choisi pour utilisation avec la classe **#OPEN-TYPE** est toujours le même que l'ensemble d'objets de codage combinés appliqué à la classe **#OPEN-TYPE** du fait que ces objets de codage ne contiennent pas un ENCODED WITH (voir § 13.2.10.5 et 13.2.9 d)).

7 Paragraphe 21.11.1

Remplacer le § 21.11.1 par le suivant:

21.11.1 Le type "RangeCondition" est le suivant:

```
RangeCondition ::= ENUMERATED
    {unbounded-or-no-lower-bound,
     semi-bounded-with-negatives,
     bounded-with-negatives,
     semi-bounded-without-negatives,
     bounded-without-negatives,
     test-lower-bound,
     test-upper-bound,
     test-range}
```

8 Paragraphe 21.11.4

Remplacer le § 21.11.4 par le suivant:

21.11.4 Le prédicat est satisfait pour ~~chaque~~ chacune des cinq premières valeurs d'énumération du § 21.11.1 si et seulement si les conditions suivantes sont satisfaites par les limites relatives à la classe de codage dans la catégorie des entiers:

- a) **unbounded-or-no-lower-bound**: soit il n'y a pas de limites, ou il y a seulement une limite supérieure mais aucune limite inférieure.
- b) **semi-bounded-with-negatives**: il y a une limite inférieure qui est inférieure à zéro, mais aucune limite supérieure.
- c) **bounded-with-negatives**: il y a une limite inférieure qui est inférieure à zéro et une limite supérieure.
- d) **semi-bounded-without-negatives**: il y a une limite inférieure qui est supérieure ou égale à zéro, mais aucune limite supérieure.
- e) **bounded-without-negatives**: il y a une limite inférieure qui est supérieure ou égale à zéro, et une limite supérieure.

NOTE – Pour tout ensemble donné de limites, exactement un seul prédicat sera satisfait.

9 Paragraphe 21.11.5

Ajouter le nouveau § 21.11.5 suivant:

21.11.5 Si les trois dernières valeurs d'énumération du § 21.11.1 sont utilisées, une valeur du type "**Comparison**" (voir § 21.11 *bis*) doit être fournie, ainsi qu'une valeur **comparator** entière. Si les autres valeurs d'énumération sont utilisées, les valeurs précitées ne doivent pas être fournies.

10 Nouveau paragraphe 21.11 *bis*

Ajouter, après le § 21.11, le nouveau § 21.11 *bis* suivant, et l'ajouter à la table des matières:

21.11 *bis* Le type Comparison

21.11 *bis.1* Le type "**Comparison**" est le suivant:

```
Comparison ::= ENUMERATED
    {equal-to,
     not-equal-to,
     greater-than,
     less-than,
     greater-than-or-equal-to,
     less-than-or-equal-to}
```

21.11 *bis.2* Il n'y a pas de valeur par défaut pour une propriété de codage de ce type.

21.11 *bis.3* Une propriété de codage de type "**Comparison**" est utilisée pour vérifier une propriété identifiée d'une classe par rapport à une valeur entière (le **comparator**).

21.11 *bis.4* Le prédicat utilisant une comparaison "**Comparison**" est satisfait pour chaque valeur d'énumération si et seulement si la propriété identifiée satisfait aux conditions suivantes:

- a) **equal-to**: sa valeur est égale à celle de la valeur **comparator** entière spécifiée.
- b) **not-equal-to**: sa valeur est différente de celle de la valeur **comparator** entière spécifiée.
- c) **greater-than**: sa valeur est supérieure à celle de la valeur **comparator** entière spécifiée.
- d) **less-than**: sa valeur est inférieure à celle de la valeur **comparator** entière spécifiée.
- e) **greater-than-or-equal-to**: sa valeur est supérieure ou égale à celle de la valeur **comparator** entière spécifiée.
- f) **less-than-or-equal-to**: sa valeur est inférieure ou égale à celle de la valeur **comparator** entière spécifiée.

11 Paragraphe 21.12.1

Remplacer le § 21.12.1 par le suivant:

21.12.1 Le type "SizeRangeCondition" est le suivant:

```
SizeRangeCondition ::= ENUMERATED
    {no-ub-with-zero-lb,
     ub-with-zero-lb,
     no-ub-with-non-zero-lb,
     ub-with-non-zero-lb,
     fixed-size,
     test-lower-bound,
     test-upper-bound,
     test-range}
```

12 Paragraphe 21.12.4

Remplacer le § 21.12.4 par le suivant:

21.12.4 Le prédicat est satisfait pour ~~chaque~~ chacune des cinq premières valeurs d'énumération du § 21.12.1 si et seulement si la contrainte de longueur effective répond aux conditions suivantes:

- a) **no-ub-with-zero-lb**: il n'y a pas de limite supérieure sur la taille et la limite inférieure est zéro.
- b) **ub-with-zero-lb**: il y a une limite supérieure sur la taille et la limite inférieure est zéro.
- c) **no-ub-with-non-zero-lb**: il n'y a pas de limite supérieure sur la taille et la limite inférieure est différente de zéro.
- d) **ub-with-non-zero-lb**: il y a une limite supérieure sur la taille et la limite inférieure est différente de zéro.
- e) **fixed-size**: la limite inférieure et la limite supérieure sur la taille ont la même valeur.

NOTE – Seul le cas de la valeur "**fixed-size**" recouvre d'autres prédicats.

13 Nouveau paragraphe 21.12.5

Ajouter, après le 21.12.4, le nouveau § 21.12.5 suivant:

21.12.5 Si les trois dernières valeurs d'énumération du § 21.12.1 sont utilisées, une valeur du type "**Comparison**" (voir § 21.11 *bis*) doit être fournie, ainsi qu'une valeur **comparator** entière. Si les autres valeurs d'énumération sont utilisées, les valeurs précitées ne doivent pas être fournies.

14 Nouveau paragraphe 21.16

Ajouter le nouveau § 21.16 suivant:

21.16 Le type IntegerMapping

21.16.1 Le type "IntegerMapping" est le suivant:

```
IntegerMapping ::= SET OF SEQUENCE {
    source SET OF INTEGER,
    result INTEGER} (CONSTRAINED BY { /* l'intersection des composants
                                         de la source doit être vide */ })
```

21.16.2 Le type "IntegerMapping" est utilisé pour spécifier explicitement une transformée "ints-to-ints".

15 Paragraphe 23.2.3.8

Remplacer le § 23.2.3.8 par le suivant:

23.2.3.8 Si un objet de codage contenu dans la liste ordonnée "**REPETITION-ENCODINGS**" est défini au moyen de "**IF**" ou "**IF-ALL**", dans ce cas tous les objets de codage précédents de cette liste doivent être définis au moyen de "**IF**" ou "**IF-ALL**".

16 Paragraphe 23.4.3.8

Remplacer le § 23.4.3.8 par le suivant:

23.4.3.8 Si un objet de codage contenu dans la liste ordonnée de "REPETITION-ENCODINGS" est défini au moyen de "IF" ou "IF-ALL", dans ce cas tous les objets de codage précédents de cette liste doivent être définis au moyen de "IF" ou "IF-ALL".

17 Paragraphe 23.6.2.3

Remplacer le § 23.6.2.3 par le suivant:

23.6.2.3 Si un objet de codage contenu dans la liste ordonnée de la spécification "ENCODINGS" est défini au moyen de "IF" ou "IF-ALL", dans ce cas tous les objets de codage précédents de cette liste doivent être définis au moyen de "IF" ou "IF-ALL".

18 Paragraphe 23.7.1

Remplacer le § 23.7.1 par le suivant:

23.7.1 La syntaxe définie

La syntaxe de définition des objets de codage pour la classe #CONDITIONAL-INT est définie comme suit:

```
#CONDITIONAL-INT ::= ENCODING-CLASS {

    -- Condition (voir § 21.11)
    &range-condition           RangeCondition OPTIONAL,
    &comparison                Comparison OPTIONAL,
    &comparator                INTEGER OPTIONAL,
    &Range-conditions          RangeCondition ORDERED OPTIONAL,
    &Comparisons               Comparison ORDERED OPTIONAL,
    &Comparators              INTEGER ORDERED OPTIONAL,

    -- Spécification de remplacement limité aux structures (voir § 22.1)
    &#Replacement-structure    OPTIONAL,
    &replacement-structure-encoding-object &#Replacement-structure OPTIONAL,

    -- Spécification de préalignement et de bourrage (voir § 22.2)
    &encoding-space-pre-alignment-unit Unit (ALL EXCEPT repetitions)
                                DEFAULT bit,
    &encoding-space-pre-padding Padding DEFAULT zero,
    &encoding-space-pre-pattern Non-Null-Pattern (ALL EXCEPT
                                different:any) DEFAULT bits:'0'B,

    -- Spécification du pointeur de début (voir § 22.3)
    &start-pointer             REFERENCE OPTIONAL,
    &start-pointer-unit        Unit (ALL EXCEPT repetitions)
                                DEFAULT bit,
    &Start-pointer-encoder-transforms #TRANSFORM ORDERED OPTIONAL,

    -- Spécification de l'espace de codage (voir § 22.4)
    &encoding-space-size       EncodingSpaceSize
                                DEFAULT self-delimiting-values,
    &encoding-space-unit       Unit (ALL EXCEPT repetitions)
                                DEFAULT bit,
    &encoding-space-determination EncodingSpaceDetermination
                                DEFAULT field-to-be-set,
    &encoding-space-reference   REFERENCE OPTIONAL,
    &Encoder-transforms         #TRANSFORM ORDERED OPTIONAL,
    &Decoder-transforms         #TRANSFORM ORDERED OPTIONAL,

    -- Codage de valeur
    &Transform                 #TRANSFORM ORDERED OPTIONAL,
    &encoding                  ENUMERATED
                                {positive-int, twos-complement,
                                reverse-positive-int,
                                reverse-twos-complement}
```

```

                                DEFAULT twos-complement,

-- Bourrage et justification de valeur (voir § 22.8)
&value-justification           Justification DEFAULT right:0,
&value-pre-padding            Padding DEFAULT zero,
&value-pre-pattern            Non-Null-Pattern DEFAULT bits:'0'B,
&value-post-padding           Padding DEFAULT zero,
&value-post-pattern           Non-Null-Pattern DEFAULT bits:'0'B,
&unused-bits-determination    UnusedBitsDetermination
                                DEFAULT field-to-be-set,

&unused-bits-reference        REFERENCE OPTIONAL,
&Unused-bits-encoder-transforms #TRANSFORM ORDERED OPTIONAL,
&Unused-bits-decoder-transforms #TRANSFORM ORDERED OPTIONAL,

-- Spécification de pointeur d'identification (voir § 22.9)
&exhibited-handle             PrintableString OPTIONAL,
&Handle-positions             INTEGER (0..MAX) OPTIONAL,
&Handle-value                 HandleValue DEFAULT tag:any,

-- Spécification de l'inversion de l'ordre des bits (voir § 22.12)
&bit-reversal                 ReversalSpecification
                                DEFAULT no-reversal
}
WITH SYNTAX {
[IF &range-condition [&comparison &comparator]]
[IF-ALL &Range-conditions [&Comparisons &Comparators]]
[ELSE]
[REPLACE
    [STRUCTURE]
    WITH &#Replacement-structure
        [ENCODED BY &replacement-structure-encoding-object]]
[ALIGNED TO
    [NEXT]
    [ANY]
    &encoding-space-pre-alignment-unit
    [PADDING &encoding-space-pre-padding
    [PATTERN &encoding-space-pre-pattern]]]
[START-POINTER &start-pointer
    [MULTIPLE OF &start-pointer-unit]
    [ENCODER-TRANSFORMS &Start-pointer-encoder-transforms]]
ENCODING-SPACE
    [SIZE &encoding-space-size
        [MULTIPLE OF &encoding-space-unit]]
    [DETERMINED BY &encoding-space-determination]
    [USING &encoding-space-reference
        [ENCODER-TRANSFORMS &Encoder-transforms]
        [DECODER-TRANSFORMS &Decoder-transforms]]
[TRANSFORMS &Transforms]
[ENCODING &encoding]
[VALUE-PADDING
    [JUSTIFIED &value-justification]
    [PRE-PADDING &value-pre-padding
        [PATTERN &value-pre-pattern]]
    [POST-PADDING &value-post-padding
        [PATTERN &value-post-pattern]]
    [UNUSED BITS
        [DETERMINED BY &unused-bits-determination]
        [USING &unused-bits-reference
            [ENCODER-TRANSFORMS &Unused-bits-encoder-transforms]
            [DECODER-TRANSFORMS &Unused-bits-decoder-transforms]]]]
[EXHIBITS HANDLE &exhibited-handle AT &Handle-positions
    [AS &handle-value]]
[BIT-REVERSAL &bit-reversal]
}

```

19 Paragraphe 23.7.2.2

Remplacer le § 23.7.2.2 par le suivant:

23.7.2.2 La syntaxe autorise la spécification d'une condition unique concernant les limites de l'entier pour que ce codage soient appliqué (utilisation de "IF"). Elle autorise également la spécification que toutes les conditions d'un ensemble de conditions soient remplies (utilisation de "IF-ALL"). Elle autorise également la spécification qu'il n'y a pas de condition. L'utilisation de "ELSE", ou l'omission d'~~aussi bien de~~ "IF" ~~que~~, de "IF-ALL" et de "ELSE" spécifie qu'il n'y a pas de condition. "IF-ALL" doit être utilisé avec trois listes si une ou plusieurs des conditions de catégorie de longueur nécessitent une comparaison, et doit être utilisé avec une liste autrement. En cas d'utilisation de trois listes, les conditions de catégorie de longueur qui ne nécessitent pas une comparaison ou un comparateur (le cas échéant) doivent suivre toutes les conditions qui nécessitent une comparaison, et ne doivent comporter aucune entrée correspondante dans les deuxième et troisième listes. En utilisant "IF-ALL" avec trois listes, les listes doivent être interprétées comme une liste de prédicats utilisant les valeurs occupant les positions correspondantes dans les trois listes.

NOTE – Il est recommandé que les trois listes soient formatées de manière à fournir une condition dans chaque colonne.

EXEMPLE:

```
IF-ALL {test-lower-bound, test-range , bounded-with-negatives }
      {greater-than , less-than-or-equal-to }
      {-10 , 20 }
```

20 Paragraphe 23.7.2.4

Remplacer le § 23.7.2.4 par le suivant:

23.7.2.4 Au plus un des mots "IF", "IF-ALL" ou "ELSE" doit être présent.

21 Paragraphes 23.7.2.6, 23.7.2.7 et 23.7.2.8

Remplacer les § 23.7.2.6, 23.7.2.7 et 23.7.2.8 par les suivants:

23.7.2.6 Il y a erreur de spécification ou d'application ECN si une éventuelle transformée contenue dans la spécification "TRANSFORMS" n'est pas réversible pour la valeur abstraite à laquelle elle est appliquée. La première transformée de "TRANSFORMS", si présente, doit avoir une source qui est un entier et la dernière transformée doit avoir un résultat qui est un entier.

NOTE – Les essais de ~~las~~ conditions "IF" et "IF-ALL" ~~a-ont~~ lieu aux limites de la valeur originale, et ~~n'est~~ ~~ne sont~~ pas affectés par ces transformées.

23.7.2.7 La transformée "INT-TO-INT" ayant la valeur "subtract:lower-bound" ne doit être incluse que si la condition "IF" ou "IF-ALL" restreint l'application de ce codage aux classes de la catégorie des entiers avec une limite inférieure, et (si présente) doit être la première transformée dans la liste.

23.7.2.8 La production "ENCODING-SPACE SIZE" ne doit pas être à la valeur "fixed-to-max" à moins que la condition "IF" ou "IF-ALL" ne restreigne le codage à une classe ayant aussi bien une limite supérieure qu'une limite inférieure.

22 Paragraphe 23.9.3.8

Remplacer le § 23.9.3.8 par le suivant:

23.9.3.8 Si un objet de codage contenu dans la liste ordonnée "REPETITION-ENCODINGS" est défini au moyen de "IF" ou "IF-ALL", dans ce cas tous les objets de codage précédents de cette liste doivent être définis au moyen de "IF" ou "IF-ALL".

23 Nouveau paragraphe 23.9 bis

Ajouter, après le § 23.9, le nouveau § 23.9 bis et actualiser en conséquence la table des matières:

23.9 bis Définition des objets de codage pour les classes de la catégorie des types ouverts**23.9 bis.1 La syntaxe définie**

La syntaxe de définition des objets de codage pour les classes de la catégorie des types ouverts est définie comme suit:

```
#OPEN-TYPE ::= ENCODING-CLASS {

    -- Spécification de remplacement limité aux structures (voir § 22.1)
    &#Replacement-structure          OPTIONAL,
    &replacement-structure-encoding-object
                                     &#Replacement-structure OPTIONAL,

    -- Spécification de préalignement et de bourrage (voir § 22.2)
    &encoding-space-pre-alignment-unit
                                     Unit (ALL EXCEPT repetitions)
                                     DEFAULT bit,

    &encoding-space-pre-padding       Padding DEFAULT zero,
    &encoding-space-pre-pattern       Non-Null-Pattern (ALL EXCEPT different:any)
                                     DEFAULT bits:'0'B,

    -- Spécification du pointeur de début (voir § 22.3)
    &start-pointer                    REFERENCE OPTIONAL,
    &start-pointer-unit               Unit (ALL EXCEPT repetitions) DEFAULT bit,
    &Start-pointer-encoder-transforms #TRANSFORM ORDERED OPTIONAL,

    -- Spécification de l'espace de codage (voir § 22.4)
    &encoding-space-size              EncodingSpaceSize
                                     DEFAULT self-delimiting-values,
    &encoding-space-unit              Unit (ALL EXCEPT repetitions)
                                     DEFAULT bit,
    &encoding-space-determination     EncodingSpaceDetermination
                                     DEFAULT field-to-be-set,
    &encoding-space-reference         REFERENCE OPTIONAL,
    &Encoder-transforms               #TRANSFORM ORDERED OPTIONAL,
    &Decoder-transforms               #TRANSFORM ORDERED OPTIONAL,

    -- Codage des types ouverts
    &Known-structure-encodings        #ENCODINGS OPTIONAL,
    &Unknown-structure                OPTIONAL,
    &Unknown-structure-encodings      #ENCODINGS OPTIONAL,

    -- Bourrage et justification de valeur (voir § 22.8)
    &value-justification              Justification DEFAULT right:0,
    &value-pre-padding                Padding DEFAULT zero,
    &value-pre-pattern                Non-Null-Pattern DEFAULT bits:'0'B,
    &value-post-padding               Padding DEFAULT zero,
    &value-post-pattern               Non-Null-Pattern DEFAULT bits:'0'B,
    &unused-bits-determination        UnusedBitsDetermination
                                     DEFAULT field-to-be-set,
    &unused-bits-reference            REFERENCE OPTIONAL,
    &Unused-bits-encoder-transforms   #TRANSFORM ORDERED OPTIONAL,
    &Unused-bits-decoder-transforms   #TRANSFORM ORDERED OPTIONAL,

    -- Spécification de l'inversion de l'ordre des bits (voir § 22.12)
    &bit-reversal                     ReversalSpecification
                                     DEFAULT no-reversal
}
WITH SYNTAX {
    [REPLACE
        [STRUCTURE]
        WITH &#Replacement-structure
            [ENCODED BY &replacement-structure-encoding-object]]
    [ALIGNED TO
        [NEXT]
        [ANY]
        &encoding-space-pre-alignment-unit
        [PADDING &encoding-space-pre-padding
        [PATTERN &encoding-space-pre-pattern]]]
```

```

[START-POINTER &start-pointer
  [MULTIPLE OF &start-pointer-unit]
  [ENCODER-TRANSFORMS &Start-pointer-encoder-transforms]]
ENCODING-SPACE
  [SIZE &encoding-space-size
    [MULTIPLE OF &encoding-space-unit]]
  [DETERMINED BY &encoding-space-determination]
  [USING &encoding-space-reference
    [ENCODER-TRANSFORMS &Encoder-transforms]
    [DECODER-TRANSFORMS &Decoder-transforms]]
  [ENCODED WITH &Known-structure-encodings]
  [UNKNOWN IS &Unknown-structure
    [ENCODED WITH &Unknown-structure-encodings]]
  [VALUE-PADDING
    [JUSTIFIED &value-justification]
    [PRE-PADDING &value-pre-padding
      [PATTERN &value-pre-pattern]]
    [POST-PADDING &value-post-padding
      [PATTERN &value-post-pattern]]
    [UNUSED BITS
      [DETERMINED BY &unused-bits-determination]
      [USING &unused-bits-reference
        [ENCODER-TRANSFORMS &Unused-bits-encoder-transforms]
        [DECODER-TRANSFORMS &Unused-bits-decoder-transforms]]]]
  [EXHIBITS HANDLE &exhibited-handle AT &Handle-positions
    [AS &handle-value]]
  [BIT-REVERSAL &bit-reversal]
}

```

23.9 bis.2 Modèle de codage de classes de la catégorie des types ouverts

23.9 bis.2.1 Le modèle de codage des types ouverts est le suivant:

- La classe de la catégorie des types ouverts peut être remplacée par une autre structure pour assurer la délimitation des longueurs, si nécessaire.
- L'objet de codage défini pour cette catégorie applique l'ensemble d'objets de codage de la construction "ENCODED WITH" au type dont la valeur doit être codée pour le type ouvert. En l'absence de construction "ENCODED WITH", on utilise dans ce cas l'ensemble d'objets de codage combinés en cours.
- Le décodeur demandera à l'application l'identification du type codé dans le type ouvert. En réponse, l'application communiquera l'identification du type, qui sera ensuite décodé, ou indiquera que le type codé dans le type ouvert est impossible à déterminer (en envoyant un message "type inconnu").
- Si la réponse est "inconnu" et que la structure "UNKNOWN IS" soit présente, dans ce cas le décodeur utilisera la structure "UNKNOWN IS" ainsi que la construction "ENCODED WITH" figurant dans la structure "UNKNOWN IS" (si elle est présente), pour déterminer la fin de l'espace de codage.
- Si la réponse est "inconnu" et que la structure "UNKNOWN IS" soit absente, dans ce cas la taille de l'espace de codage peut être déterminée par la production "ENCODING-SPACE" (voir § 23.9 bis.3.3), et le décodeur renverra à l'application tous les bits contenus dans l'espace de codage défini à l'exception des bits de prébourrage et de postbourrage de valeur.

23.9 bis.2.2 En cas de décodage inconnu, le décodeur transmettra les bits constituant le codage inconnu à l'application sous la forme de la valeur du type ouvert.

23.9 bis.3 Finalité et restrictions

23.9 bis.3.1 Cette syntaxe est utilisée afin de définir la manière dont un type ouvert est codé, ainsi que les moyens qu'un décodeur utilise pour déterminer la fin du codage d'un type inconnu figurant dans un type ouvert.

23.9 bis.3.2 Si "REPLACE STRUCTURE" est activé, aucun autre paramètre ne doit être activé.

23.9 bis.3.3 Si "ENCODING-SPACE SIZE" est "self-delimiting" (autodélimitateur), dans ce cas la structure "UNKNOWN IS" doit être activée.

23.9 bis.4 Actions du codeur

23.9 bis.4.1 Pour tout groupe de propriétés de codage qui est activé, le codeur doit exécuter les actions de codage spécifiées au § 22, dans l'ordre suivant et conformément à la définition d'objet de codage:

- remplacement;

- b) préalignement et bourrage;
- c) pointeur de début;
- d) espace de codage (voir § 23.9 bis.4.3);
- e) codage de type ouvert (voir § 23.9 bis.4.2);
- f) bourrage et justification de valeur (voir § 23.9 bis.4.5);
- g) inversion de l'ordre des bits.

23.9 bis.4.2 Le codeur doit coder la valeur du type fourni par l'application en utilisant l'ensemble d'objets de codage "ENCODED WITH" si celui-ci est présent; autrement, l'ensemble d'objets de codage combinés en cours doit être utilisé.

23.9 bis.4.3 Si "ENCODING-SPACE SIZE" est "variable-with-determinant" ou "encoder-option-with-determinant", dans ce cas l'espace de codage doit correspondre au nombre minimal de "MULTIPLE OF" unités nécessaires pour contenir la configuration ("s", par exemple), sous réserve des dispositions du § 23.9 bis.4.5.

23.9 bis.4.4 Un codeur (à titre d'option de codeur) peut augmenter "s" (telle que déterminée au § 23.9 bis.4.3) de "MULTIPLE OF" unités (sous réserve d'éventuelles restrictions imposées par l'étendue de valeurs d'un quelconque champ "added-field" ou "asn1-field") si la production "ENCODING-SPACE SIZE" est activée à la valeur "encoder-option-with-determinant".

23.9 bis.4.5 Si le nombre de bits inutilisés est différent de zéro, dans ce cas la justification de valeur "VALUE-JUSTIFICATION" doit être appliquée en utilisant les valeurs activées ou les valeurs par défaut.

23.9 bis.5 Actions du décodeur

23.9 bis.5.1 Pour tout groupe de propriétés de codage qui est activé, le décodeur doit exécuter les actions de décodage spécifiées au § 22, dans l'ordre suivant et conformément à la définition d'objet de codage:

- a) préalignement et bourrage;
- b) pointeur de début;
- c) espace de codage;
- d) inversion de l'ordre des bits;
- e) bourrage et justification de valeur;
- f) décodage de type ouvert (voir § 23.9 bis.5.2).

23.9 bis.5.2 Pour le décodage d'un type ouvert, le décodeur doit demander à l'application de lui indiquer le type qui a été codé et doit décoder une valeur de ce type ou de la structure "UNKNOWN IS" conformément aux spécifications "ENCODED WITH" figurant dans la structure "UNKNOWN IS".

23.9 bis.5.3 Si le codage était d'un type inconnu, les bits constituant le codage inconnu (en l'absence de bits de prébourrage et, le cas échéant, de bits de prébourrage et de postbourrage de valeur) doivent être transmis à l'application sous la forme de la valeur du type ouvert.

24 Paragraphe 23.12.2.3

Remplacer le § 23.12.2.3 par le suivant:

23.12.2.3 Si un objet de codage dans la liste ordonnée "REPETITION-ENCODINGS" est défini au moyen de "IF" ou "IF-ALL", dans ce cas tous les objets de codage précédents de cette liste doivent être définis au moyen de "IF" ou "IF-ALL".

25 Paragraphe 23.13.1

Remplacer le § 23.13.1 par le suivant:

23.13.1 La syntaxe définie

La syntaxe de définition des objets de codage pour la classe #CONDITIONAL-REPETITION est définie comme suit:

```
#CONDITIONAL-REPETITION ::= ENCODING-CLASS {
    -- Condition (voir § 21.12)
    &size-range-condition          SizeRangeCondition OPTIONAL,
```

```

&comparison Comparison OPTIONAL,
&comparator INTEGER OPTIONAL,
&Size-range-conditions SizeRangeCondition ORDERED OPTIONAL,
&Comparisons Comparison ORDERED OPTIONAL,
&Comparators INTEGER ORDERED OPTIONAL,

-- Spécification de remplacement de structure ou de composant (voir § 22.1)
&#Replacement-structure OPTIONAL,
&replacement-structure-encoding-object &#Replacement-structure OPTIONAL,
&#Head-end-structure OPTIONAL,

-- Spécification de préalignement et de bourrage (voir § 22.2)
&encoding-space-pre-alignment-unit Unit (ALL EXCEPT repetitions) DEFAULT bit,
&encoding-space-pre-padding Padding DEFAULT zero,
&encoding-space-pre-pattern Non-Null-Pattern (ALL EXCEPT different:any)
DEFAULT bits:'0'B,

-- Spécification du pointeur de début (voir § 22.3)
&start-pointer REFERENCE OPTIONAL,
&start-pointer-unit Unit (ALL EXCEPT repetitions) DEFAULT bit,
&Start-pointer-encoder-transforms #TRANSFORM ORDERED OPTIONAL,

-- Spécification de l'espace de répétition (voir § 22.7)
&repetition-space-size EncodingSpaceSize
DEFAULT self-delimiting-values,
&repetition-space-unit Unit DEFAULT bit,
&repetition-space-determination RepetitionSpaceDetermination
DEFAULT field-to-be-set,
&main-reference REFERENCE OPTIONAL,
&Encoder-transforms #TRANSFORM ORDERED OPTIONAL,
&Decoder-transforms #TRANSFORM ORDERED OPTIONAL,
&handle-id PrintableString
DEFAULT "default-handle",
&termination-pattern Non-Null-Pattern (ALL EXCEPT
different:any) DEFAULT '0'B,

-- Classe &repetition-alignment ENUMERATED {none, aligned}
DEFAULT none,

-- Bourrage et justification de valeur (voir § 22.8)
&value-justification Justification DEFAULT right:0,
&value-pre-padding Padding DEFAULT zero,
&value-pre-pattern Non-Null-Pattern DEFAULT bits:'0'B,
&value-post-padding Padding DEFAULT zero,
&value-post-pattern Non-Null-Pattern DEFAULT bits:'0'B,
&unused-bits-determination UnusedBitsDetermination
DEFAULT field-to-be-set,
&unused-bits-reference REFERENCE OPTIONAL,
&Unused-bits-encoder-transforms #TRANSFORM ORDERED OPTIONAL,
&unused-bits-decoder-transforms #TRANSFORM ORDERED OPTIONAL,

-- Spécification de pointeur d'identification (voir § 22.9)
&exhibited-handle PrintableString OPTIONAL,
&Handle-positions INTEGER (0..MAX) OPTIONAL,
&Handle-value HandleValue DEFAULT tag: any,

-- Spécification de l'inversion de l'ordre des bits (voir § 22.12)
&bit-reversal ReversalSpecification
DEFAULT no-reversal
}
WITH SYNTAX {
[IF &size-range-condition [&comparison &comparator]]
[IF-ALL &Size-range-conditions [&Comparisons &Comparators]]
[ELSE]
[REPLACE
[STRUCTURE]
[COMPONENT]
[ALL COMPONENTS]
WITH &Replacement-structure

```

```

[ENCODED BY &replacement-structure-encoding-object
  [INSERT AT HEAD &#Head-end-structure]]]
[ALIGNED TO
  [NEXT]
  [ANY]
  &encoding-space-pre-alignment-unit
  [PADDING &encoding-space-pre-padding
  [PATTERN &encoding-space-pre-pattern]]]
[START-POINTER &start-pointer
  [MULTIPLE OF &start-pointer-unit]
  [ENCODER-TRANSFORMS &Start-pointer-encoder-transforms]]
REPETITION-SPACE
  [SIZE &repetition-space-size
  [MULTIPLE OF &repetition-space-unit]]
  [DETERMINED BY &repetition-space-determination
  [HANDLE &handle-id]]
  [USING &main-reference
  [ENCODER-TRANSFORMS &Encoder-transforms]
  [DECODER-TRANSFORMS &Decoder-transforms]]
  [PATTERN &termination-pattern]
[ALIGNMENT &repetition-alignment]
[VALUE-PADDING
  [JUSTIFIED &value-justification]
  [PRE-PADDING &value-pre-padding
  [PATTERN &value-pre-pattern]]
  [POST-PADDING &value-post-padding
  [PATTERN &value-post-pattern]]
  [UNUSED BITS
  [DETERMINED BY &unused-bits-determination]
  [USING &unused-bits-reference
  [ENCODER-TRANSFORMS &Unused-bits-encoder-transforms]
  [DECODER-TRANSFORMS &Unused-bits-decoder-transforms]]]]
[EXHIBITS HANDLE &exhibited-handle AT &Handle-positions
[AS &handle-value]]
[BIT-REVERSAL &bit-reversal]
}

```

26 Paragraphes 23.13.2.1 et 23.13.2.2

Remplacer les § 23.13.2.1 et 23.13.2.2 par les suivants:

23.13.2.1 Cette syntaxe est utilisée afin de définir le codage d'une classe de la catégorie des répétitions sous réserve de la réalisation d'une condition sur la base des limites de la répétition (utilisation de "IF"). Elle autorise également la spécification que toutes les conditions d'un ensemble de conditions soient réalisées (utilisation de "IF-ALL"). Elle autorise également la spécification qu'il n'y a pas de condition. L'utilisation de "ELSE", ou l'omission ~~d'aussi bien de~~ "IF" ~~que~~, de "IF-ALL" et de "ELSE" spécifie qu'il n'y a pas de condition. "IF-ALL" doit être utilisé avec trois listes si une ou plusieurs des conditions de catégorie de longueur nécessitent une comparaison, et doit être utilisé avec une liste autrement. En cas d'utilisation de trois listes, les conditions de catégorie de longueur qui ne nécessitent pas une comparaison ou un comparateur (le cas échéant) doivent suivre toutes les conditions qui nécessitent une comparaison, et ne doivent comporter aucune entrée correspondante dans les deuxième et troisième listes. En utilisant "IF-ALL" avec trois listes, les listes doivent être interprétées comme une liste de prédicats utilisant les valeurs occupant les positions correspondantes dans les trois listes.

NOTE – Il est recommandé que les trois listes soient formatées de manière à fournir une condition dans chaque colonne (voir l'exemple donné au § 23.7.2.2).

23.13.2.2 Au plus un des mots "IF", "IF-ALL" et ou "ELSE" doit être présent.

27 Paragraphe 23.15

Remplacer le § 23.15 par le suivant:

23.15 Définition des objets de codage pour classes d'autres catégories

Dans cette version de la présente Recommandation | Norme internationale, il n'y a pas de syntaxe définie pour les classes appartenant aux catégories suivantes:

```

objectidentifier
open-type
real

```

28 Paragraphe 24.3.1

Remplacer le § 24.3.1 par le suivant:

24.3.1 La transformée "int-to-int" utilise la propriété de codage suivante:

&int-to-int	CHOICE	
	{increment	INTEGER (1..MAX),
	decrement	INTEGER (1..MAX),
	multiply	INTEGER (2..MAX),
	divide	INTEGER (2..MAX),
	negate	ENUMERATED{value},
	modulo	INTEGER (2..MAX),
	subtract	ENUMERATED{lower-bound},
	<u>mapping</u>	<u>IntegerMapping</u>
	} OPTIONAL	

29 Nouvea paragraphe 24.3.2 bis

Ajouter, après le § 24.3.2, le § 24.3.2bis suivant:

24.3.2 bis La définition du type utilisé dans la transformée "int-to-int" est la suivante:

```

IntegerMapping ::= SET OF SEQUENCE {
    source      SET OF INTEGER,
    result      INTEGER} (CONSTRAINED BY { /* l'intersection des composants de la
                                                source doit être vide
                                                (voir 21.16) */ })

```

30 Nouveau paragraphe 24.3.8 bis

Ajouter, après le § 24.3.8, le § 24.3.8 bis suivant:

24.3.8 bis La transformée pour la valeur "mapping:integerMapping" est définie comme suit. La valeur entière originale est remplacée par la valeur associée à l'ensemble de valeurs dont elle fait partie. Il y a erreur de spécification ECN si l'intersection des ensembles de valeurs n'est pas vide; il y a erreur d'application si l'entier original ne fait partie d'un des ensembles de valeurs.

31 Tableau 6

Ajouter à la fin du Tableau 6 le texte suivant:

mapping:integerMapping	<i>Ensembles des valeurs de la source, contenant chacune une seule valeur, les valeurs obtenues (résultats) étant différentes.</i>
-------------------------------	--

32 Paragraphe C.1

Remplacer, dans le § C.1, la production "Governor" par le texte suivant:

```

Governor ::=
    EncodingClassFieldType
    | REFERENCE
    | DefinedOrBuiltinEncodingClass
    | #ENCODINGS
    | Type

```

33 Paragraphe C.4

Remplacer le § C.4 par le texte suivant:

C.4 Liste des paramètres réels

La Rec. UIT-T X.683 | ISO/CEI 8824-4, § 9.5, est modifiée comme suit:

9.5 La production "ActualParameterList" est la suivante:

```

ActualParameterList ::=
  "{" <" ActualParameter " "," + ">"
ActualParameter ::=
  Value
  | ValueSet
  | OrderedValueList
  | DefinedOrBuiltinEncodingClass
  | EncodingObject
  | EncodingObjectSet
  | OrderedEncodingObjectList
  | identifiant
  | ComponentIdList
  | STRUCTURE
  | OUTER

```

Si le paramètre fictif correspondant est:

- a) une valeur: l'option "Value" doit être utilisée;
- b) un ensemble de valeurs: l'option "ValueSet" doit être utilisée;
- c) une liste ordonnée de valeurs de type fixe: l'option "OrderedValueList" doit être utilisée;
- d) une classe de codage: l'option "DefinedOrBuiltinEncodingClass" doit être utilisée;
- e) un objet de codage: l'option "EncodingObject" doit être utilisée;
- f) un ensemble d'objets de codage: l'option "EncodingObjectSet" doit être utilisée;
- g) une liste ordonnée d'objets de codage: l'option "OrderedEncodingObjectList" doit être utilisée;
- h) une référence: l'option "~~identifiant~~ComponentIdList", **STRUCTURE** ou **OUTER** doit être utilisée.

L'option **STRUCTURE** ne doit être ~~sélectionnée-utilisée~~ que lorsque le paramètre réel est utilisé comme spécifié au § 17.5.15. L'option **OUTER** peut être utilisée chaque fois qu'une référence est requise afin de désigner un conteneur, et identifie le conteneur du codage entier.

34 Paragraphe G.2.4

Remplacer, dans le § G.2.4, la production "Governor" par le texte suivant:

```

Governor ::=
  EncodingClassFieldType
  | REFERENCE
  | DefinedOrBuiltinEncodingClass
  | #ENCODINGS
  | Type

```


SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
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	Réseaux câblés et 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	Gestion des télécommunications y compris le RGT et maintenance des réseaux
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 de données, communication entre systèmes ouverts et sécurité
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de nouvelle génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

