

Remplacée par une version plus récente



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

X.690

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(07/94)

**RÉSEAUX POUR DONNÉES ET INTERCONNEXION
DES SYSTÈMES OUVERTS**

**RÉSEAUTAGE OSI ET ASPECTS
DES SYSTÈMES – NOTATION DE SYNTAXE
ABSTRAITE NUMÉRO UN (ASN.1)**

**TECHNOLOGIES DE L'INFORMATION –
RÈGLES DE CODAGE DE LA NOTATION
DE SYNTAXE ABSTRAITE NUMÉRO UN:
SPÉCIFICATION DES RÈGLES DE CODAGE
DE BASE, DES RÈGLES DE CODAGE
CANONIQUES ET DES RÈGLES DE CODAGE
DISTINCTIVES**

Recommandation UIT-T X.690

Remplacée par une version plus récente

(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.690 de l'UIT-T a été approuvé le 1^{er} juillet 1994. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 8825-1.

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

| | <i>Page</i> |
|------|--|
| 1 | Domaine d'application..... 1 |
| 2 | Références normatives 1 |
| 2.1 | Recommandations et Normes internationales identiques..... 1 |
| 2.2 | Autres références 2 |
| 3 | Définitions..... 2 |
| 4 | Abréviations 3 |
| 5 | Notation..... 3 |
| 6 | Conventions..... 3 |
| 7 | Conformité 3 |
| 8 | Règles de codage de base 3 |
| 8.1 | Règles générales de codage..... 3 |
| 8.2 | Codage d'une valeur booléenne 7 |
| 8.3 | Codage d'une valeur entière 8 |
| 8.4 | Codage d'une valeur énumérée 8 |
| 8.5 | Codage d'une valeur réelle 8 |
| 8.6 | Codage d'une valeur de type chaîne binaire 10 |
| 8.7 | Codage d'une valeur de type chaîne d'octets 11 |
| 8.8 | Codage d'une valeur vide 11 |
| 8.9 | Codage d'une valeur de type séquence..... 12 |
| 8.10 | Codage d'une valeur de type séquence-de..... 12 |
| 8.11 | Codage d'une valeur de type ensemble 12 |
| 8.12 | Codage d'une valeur de type ensemble-de 12 |
| 8.13 | Codage d'une valeur de type choix 13 |
| 8.14 | Codage d'une valeur étiquetée..... 13 |
| 8.15 | Codage d'une valeur de type ouvert 13 |
| 8.16 | Codage d'une valeur de type instance-de 14 |
| 8.17 | Codage d'une valeur de type valeur de donnée de présentation encapsulée 14 |
| 8.18 | Codage d'une valeur de type externe..... 15 |
| 8.19 | Codage d'une valeur d'identificateur d'objet 16 |
| 8.20 | Codage d'une valeur de type chaîne de caractères avec restriction 17 |
| 8.21 | Codage d'une valeur de type chaîne de caractères sans restriction 19 |
| 9 | Règles de codage canoniques 20 |
| 9.1 | Formes de longueur..... 20 |
| 9.2 | Formes de codage des chaînes 20 |
| 9.3 | Éléments d'ensemble..... 20 |
| 10 | Règles de codage distinctives..... 21 |
| 10.1 | Formes de longueur..... 21 |
| 10.2 | Formes de codage des chaînes 21 |
| 10.3 | Éléments d'ensemble..... 21 |

Remplacée par une version plus récente

Page

| | | |
|------|--|----|
| 11 | Restrictions aux règles de codage de base applicables aux règles de codage canoniques et distinctives | 21 |
| 11.1 | Valeurs booléennes | 21 |
| 11.2 | Bits inutilisés | 21 |
| 11.3 | Valeurs réelles | 21 |
| 11.4 | Valeurs du type chaîne générale GeneralString | 22 |
| 11.5 | Éléments d'ensemble et éléments de séquence avec valeur par défaut | 22 |
| 11.6 | Éléments d'ensemble-de | 22 |
| 11.7 | Temps généralisé | 22 |
| 12 | Utilisation des règles de codage canoniques, distinctives et de base dans une définition de syntaxe de transfert | 23 |
| | Annexe A – Exemples de codages | 24 |
| A.1 | Description ASN.1 de la structure de l'enregistrement | 24 |
| A.2 | Description ASN.1 d'une valeur d'enregistrement | 24 |
| A.3 | Représentation de la valeur de cet enregistrement | 24 |
| | Annexe B – Affectation des valeurs d'identificateur d'objet | 26 |
| | Annexe C – Illustration du codage d'une valeur réelle | 27 |
| | Annexe D – Utilisation des règles de codage distinctives (DER) et canoniques (CER) en authentification d'origine des données | 29 |
| D.1 | Problème à résoudre | 29 |
| D.2 | Approche de la solution | 30 |
| D.3 | Optimisation du produit | 30 |

Remplacée par une version plus récente

Résumé

La présente Recommandation | Norme internationale définit un ensemble de règles de codage de base (BER) applicables aux valeurs des types définis au moyen de la notation ASN.1. L'application de ces règles de codage produit une syntaxe de transfert pour de telles valeurs. Il est implicitement entendu que ces règles de codage servent également au décodage. La présente Recommandation | Norme internationale définit également un ensemble de règles de codage distinctives (DER) et un ensemble de règles de codage canoniques (CER) qui permettent tous deux de déclarer des contraintes sur les règles de codage de base (BER). La principale différence entre ces deux ensembles de règles est que les DER utilisent des formes de codage de longueur définie alors que les CER utilisent les formes de longueur indéfinie. Les DER sont mieux adaptées au codage des petites valeurs, et les CER à celui des grandes valeurs. Il est implicitement entendu que ces règles de codage servent également au décodage.

Remplacée par une version plus récente

Introduction

Les Rec. UIT-T X.680 | ISO/CEI 8824-1, UIT-T X.681 | ISO/CEI 8824-2, UIT-T X.682 | ISO/CEI 8824-3, UIT-T X.683 | ISO/CEI 8824-4, (Syntaxe abstraite numéro un ou ASN.1) spécifient une notation de définition de syntaxes abstraites, permettant aux normes de la couche application de définir les types d'informations nécessaires au transfert des données au moyen du service de présentation. Elles définissent également une notation pour la spécification des valeurs de chaque type défini.

La présente Recommandation | Norme internationale définit les règles de codage applicables aux valeurs des types définis au moyen de la notation ASN.1. L'application de ces règles de codage produit une syntaxe de transfert pour ces valeurs. Il est implicitement entendu que la spécification de ces règles de codage s'applique également au décodage.

Plusieurs ensembles de règles de codage peuvent être appliqués aux valeurs des types définis au moyen de la notation ASN.1. La présente Recommandation | Norme internationale définit trois ensembles de règles de codage, appelés **règles de codage de base**, **règles de codage canoniques** et **règles de codage distinctives**. Alors que les règles de codage de base offrent au codeur différentes possibilités de codage pour les valeurs, les règles de codage canoniques et distinctives sélectionnent pour chaque valeur un seul codage parmi les possibilités offertes par les règles de codage de base en éliminant toutes les options laissées par celles-ci au codeur. Les règles distinctives et les règles canoniques diffèrent par la nature des restrictions qu'elles imposent aux règles de codage de base.

Les règles distinctives conviennent mieux que les règles canoniques lorsque la valeur codée est suffisamment petite pour tenir dans la mémoire disponible et lorsqu'il est nécessaire de passer rapidement certaines valeurs encapsulées. Les règles canoniques sont mieux adaptées que les règles distinctives lorsqu'il est besoin de coder des valeurs si grandes qu'elles dépassent la capacité mémoire disponible ou lorsqu'il est nécessaire de coder et de transmettre une partie d'une valeur avant que celle-ci soit disponible dans sa totalité. Les règles de codage de base sont mieux adaptées que les règles de codage canoniques ou distinctives s'il s'agit de coder une valeur du type ensemble ou ensemble-de sans s'astreindre aux restrictions que les règles canoniques et distinctives imposent. Ceci est dû au surcroît de mémoire et de calculs que ces dernières exigent afin de garantir que les valeurs de type ensemble ou ensemble-de n'ont qu'un seul codage possible.

L'Annexe A donne des exemples d'application des règles de codage de base. Elle ne fait pas partie intégrante de la présente Recommandation | Norme internationale.

L'Annexe B résume les affectations de valeurs d'identificateurs d'objets stipulées dans la présente Recommandation | Norme internationale. Elle ne fait pas partie intégrante de la présente Recommandation | Norme internationale.

L'Annexe C donne des exemples de l'application des règles de base au codage des réels. Elle ne fait pas partie intégrante de la présente Recommandation | Norme internationale.

L'Annexe D montre comment utiliser les règles de codage distinctives pour assurer un service d'intégrité pour les communications OSI. Elle ne fait pas partie intégrante de la présente Recommandation | Norme internationale.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION – RÈGLES DE CODAGE
DE LA NOTATION DE SYNTAXE ABSTRAITE NUMÉRO UN:
SPÉCIFICATION DES RÈGLES DE CODAGE DE BASE,
DES RÈGLES DE CODAGE CANONIQUES
ET DES RÈGLES DE CODAGE DISTINCTIVES**

1 Domaine d'application

La présente Recommandation | Norme internationale spécifie un ensemble de règles de codage de base qui peuvent être utilisées pour spécifier une syntaxe de transfert pour des valeurs appartenant à des types définis au moyen de la notation spécifiée dans les Rec. UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, et UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, appelées collectivement syntaxe abstraite numéro un ou ASN.1. Ces règles de codage de base s'appliquent également au décodage d'une telle syntaxe de transfert pour identifier les valeurs de données transférées. La Recommandation spécifie également un ensemble de règles canoniques et distinctives qui restreignent le codage des valeurs à une seule des possibilités autorisées par les règles de codage de base.

Ces règles de codage sont utilisées au moment de la communication (par le fournisseur du service de présentation, lorsque le contexte de présentation le requiert).

2 Références normatives

Les Recommandations et les Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher 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 de l'UIT-T en vigueur.

2.1 Recommandations et Normes internationales identiques

- Recommandation UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: Le modèle de référence de base.*
- Recommandation UIT-T X.226 (1994) | ISO/CEI 8823-1:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Protocole de présentation en mode connexion: Spécification du protocole.*
- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Technologie de l'information – Notation de syntaxe abstraite numéro un: Spécification de la notation de base.*
- Recommandation UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: Spécification des objets informationnels.*
- 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, *Technologie de l'information – Notation de syntaxe abstraite numéro un: Paramétrage des spécifications de la notation de syntaxe abstraite numéro un.*

2.2 Autres références

- ISO *Registre international des jeux de caractères codés à utiliser avec la séquence d'échappement.*
- ISO/CEI 2022:1994, *Technologies de l'information – Structure de code de caractères et techniques d'extension.*
- ISO 6093:1985, *Traitement de l'information – Représentation des valeurs numériques dans les chaînes de caractères pour l'échange d'information.*
- ISO/CEI 6429:1992, *Technologies de l'information – Fonctions de commande pour les jeux de caractères codés.*
- Recommandation X.208 du CCITT (1988), *Spécification de la syntaxe abstraite numéro un (ASN.1).*
- ISO/CEI 8824-1 à 8824-4:1990, *Technologies de l'information – Interconnexion de systèmes ouverts – Spécification de la notation de syntaxe abstraite numéro un.*
- ISO/CEI 10646-1:1993, *Technologies de l'information – Jeu universel de caractères codés à plusieurs octets – Partie 1: Architecture et table multilingue.*

3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions données par ISO 7498 et par la Rec. UIT-T X.680 | ISO/CEI 8824-1 sont utilisées et les termes suivants sont définis.

3.1 conformité dynamique: Déclaration de la nécessité pour une réalisation de se conformer au comportement prescrit par la présente Recommandation | Norme internationale au cours d'une instance de communication.

3.2 conformité statique: Déclaration de la nécessité pour une réalisation de présenter un ensemble valide de caractéristiques, parmi celles définies par la présente Recommandation | Norme internationale.

3.3 valeur de données: Information spécifiée comme valeur d'un type, le type et la valeur étant définis en ASN.1.

3.4 codage (d'une valeur de données): Séquence d'octets complète utilisée pour représenter la valeur de données.

3.5 champ d'identification: Partie du codage d'une valeur de données servant à identifier le type de la valeur.

NOTE – Certaines Recommandations UIT-T utilisent l'expression "élément de données" pour désigner cette séquence; cette expression n'est pas utilisée dans la présente Recommandation | Norme internationale, car d'autres Recommandations | Normes internationales l'utilisent au sens de "valeur de données".

3.6 champ de longueur: Partie du codage d'une valeur de données placée à la suite du champ d'identification, et servant à déterminer la longueur du codage.

3.7 champ de contenu: Partie du codage d'une valeur de données représentant une valeur particulière qui la distingue des autres valeurs du même type.

3.8 champ de fin de contenu: Partie du codage d'une valeur de données placée à sa fin et servant à indiquer la fin du codage.

NOTE – Les codages ne nécessitent pas tous des octets de fin de contenu.

3.9 codage primitif: Codage d'une valeur de donnée dans lequel le champ de contenu représente directement la valeur.

3.10 codage structuré: Codage d'une valeur de donnée dans lequel le champ de contenu est le codage complet d'une ou plusieurs autres valeurs de données.

3.11 destinataire: Réalisation décodant la séquence générée par un expéditeur pour déterminer la valeur de données qui a été codée.

3.12 expéditeur: Réalisation codant une valeur de donnée pour la transférer.

3.13 bit de fin à 0: Bit à 0 en dernière position d'une valeur de chaîne binaire.

NOTE – Le 0 d'une valeur de chaîne binaire constituée d'un bit unique de valeur nulle est un bit de fin à 0. Sa suppression transforme la chaîne en une chaîne vide.

4 Abréviations

| | |
|-------|--|
| ASN.1 | Notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>) |
| BER | Règles de codage de base (de l'ASN.1) (<i>basic encoding rules</i>) |
| CER | Règles de codage canoniques (de l'ASN.1) (<i>canonical encoding rules</i>) |
| DER | Règles de codage distinctives (de l'ASN.1) (<i>distinguished encoding rules</i>) |
| ULA | Architecture des couches supérieures (<i>upper layer architecture</i>) |

5 Notation

La présente Recommandation | Norme internationale reprend la notation définie par la Rec. UIT-T X.680 | ISO/CEI 8824-1.

6 Conventions

6.1 La présente Recommandation | Norme internationale spécifie les bits de chaque octet codé en utilisant les expressions "bit le plus significatif" et "bit le moins significatif".

NOTE – Les spécifications des couches inférieures utilisent la même notation pour définir l'ordre de transmission des bits sur une ligne série ou l'affectation des bits sur des voies parallèles.

6.2 Aux fins de la présente Recommandation | Norme internationale, les bits d'un octet sont numérotés de 8 à 1, le bit 8 étant "le plus significatif" et le bit 1 "le moins significatif".

6.3 Il est possible aux fins de la présente Recommandation | Norme internationale de comparer deux chaînes d'octets. Deux chaînes d'octets sont égales si elles ont la même longueur et si les octets de même rang sont identiques. Une chaîne d'octets S_1 est supérieure à une chaîne S_2 si et seulement si:

- soit S_1 et S_2 ont tous leurs octets de même rang égaux jusqu'à l'octet final de S_2 inclusivement, mais S_1 est plus longue que S_2 ;
- soit S_1 et S_2 diffèrent par un ou plusieurs octets de même rang, l'octet de S_1 de la première position de différence étant supérieur à son homologue de S_2 , les octets étant considérés comme des nombres binaires non signés dont le bit n est de poids 2^{n-1} .

7 Conformité

7.1 La conformité dynamique est spécifiée par les articles 8 à 12 inclusivement.

7.2 La conformité statique est définie par les normes qui spécifient l'application d'une ou plusieurs de ces règles de codage.

7.3 Les règles de base autorisent des variantes de codage sur option de l'expéditeur. Les destinataires déclarant être conformes aux règles de codage de base prendront en charge toutes les variantes possibles.

NOTE – Des exemples de ces variantes de codage figurent au 8.1.3.2 b) et au Tableau 3.

7.4 Aucune variante de codage n'est autorisée par les règles de codage canoniques et les règles de codage distinctives.

8 Règles de codage de base

8.1 Règles générales de codage

8.1.1 Structure d'un codage

8.1.1.1 Le codage d'une valeur de données comporte quatre composantes apparaissant dans l'ordre suivant:

- champ d'identification (voir 8.1.2);

Remplacée par une version plus récente ISO/CEI 8825-1 : 1995 (F)

- b) champ de longueur (voir 8.1.3);
- c) champ de contenu (voir 8.1.4);
- d) champ de fin de contenu (voir 8.1.5).

8.1.1.2 Le champ de fin de contenu ne figurera que lorsque la valeur du champ de longueur en exige la présence (voir 8.1.3).

8.1.1.3 La Figure 1 présente la structure d'un codage (primitif ou structuré). La Figure 2 présente une variante de codage structuré.

8.1.2 Champ d'identification

8.1.2.1 Le champ d'identification code l'étiquette ASN.1 (classe et numéro) du type de la valeur de données.

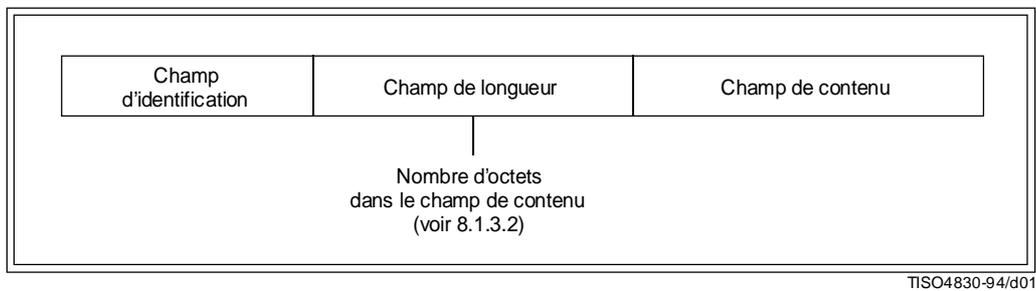


Figure 1 – Structure d'un codage

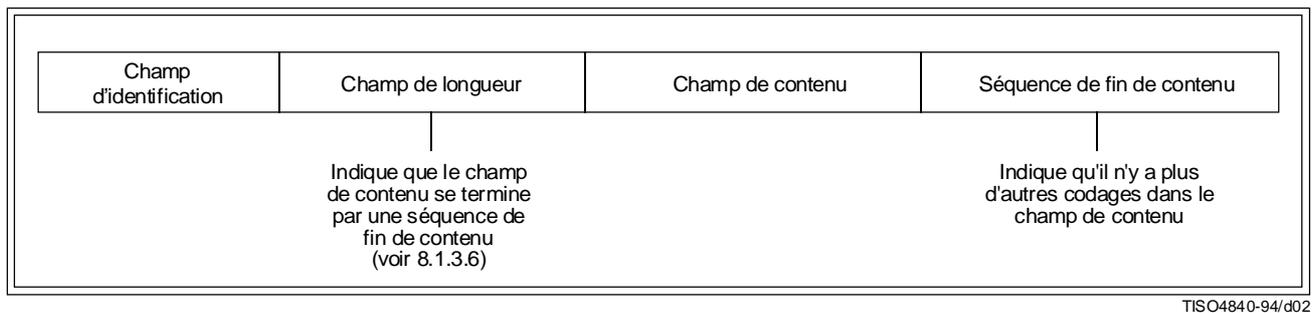


Figure 2 – Variante de codage structuré

8.1.2.2 Pour les étiquettes ayant un numéro entre 0 et 30 (inclusivement), le champ d'identification comprendra un seul octet codé comme suit:

- a) les bits 8 et 7 représentent la classe de l'étiquette conformément au Tableau 1;
- b) le bit 6 prend la valeur 0 ou 1, conformément aux règles du 8.1.2.5;
- c) les bits 5 à 1 représentent la valeur binaire du numéro de l'étiquette, le bit 5 étant le bit le plus significatif.

Tableau 1 – Codage de la classe de l'étiquette

| Classe | Bit 8 | Bit 7 |
|--------------------------|-------|-------|
| Universelle | 0 | 0 |
| Propre à une application | 0 | 1 |
| Spécifique au contexte | 1 | 0 |
| A usage privé | 1 | 1 |

8.1.2.3 La Figure 3 présente la forme du champ d'identification d'un type dont l'étiquette a un numéro compris entre 0 et 30.

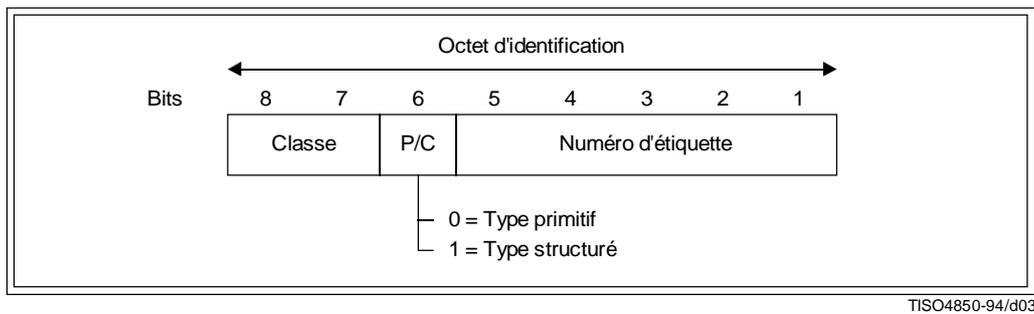


Figure 3 – Champ d'identification à un octet (étiquette de petit numéro)

8.1.2.4 Pour les étiquettes de numéro supérieur ou égal à 31, l'identificateur est composé d'un octet de tête, suivi d'un ou de plusieurs autres octets.

8.1.2.4.1 L'octet de tête est codé comme suit:

- les bits 8 et 7 représentent la classe de l'étiquette conformément au Tableau 1;
- le bit 6 prend la valeur 0 ou 1, conformément aux règles du 8.1.2.5;
- les bits 5 à 1 reçoivent la valeur 11111_2 .

8.1.2.4.2 Les octets suivants représenteront le numéro de l'étiquette codé comme suit:

- le bit 8 de chaque octet prendra la valeur 1, sauf s'il s'agit du dernier octet de l'identificateur;
- les bits 7 à 1 du premier octet suivant, suivis des bits 7 à 1 du deuxième octet à la suite, suivis à leur tour des bits 7 à 1 de chacun des octets suivants, jusques et y compris le dernier octet de l'identificateur, recevront un entier binaire non signé égal au numéro de l'étiquette, le bit 7 du premier octet étant le bit de plus fort poids;
- les bits 7 à 1 du premier octet ne doivent pas tous être à zéro.

8.1.2.4.3 La Figure 4 présente la structure du champ d'identification pour un type portant une étiquette de numéro supérieur à 30.

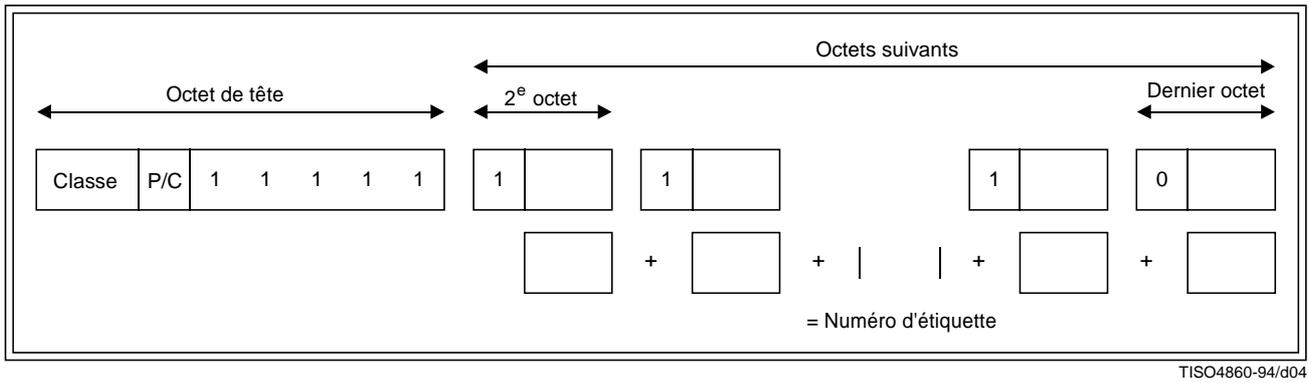


Figure 4 – Champ d'identification à plusieurs octets (étiquette de numéro élevé)

8.1.2.5 Le bit 6 sera mis à 0 si le codage est de type primitif; et à 1 s'il est de type structuré.

NOTE – Les articles suivants précisent pour chaque type si le codage est primitif ou structuré.

8.1.2.6 La Rec. UIT-T X.680 | ISO/CEI 8824-1 spécifie que l'étiquette d'un type défini au moyen du mot clé "CHOICE" prend la valeur de l'étiquette du type auquel appartient la valeur de donnée choisie.

8.1.2.7 Les paragraphes 14.2 et 14.4 de la Rec. UIT-T X.681 | ISO/CEI 8824-2 spécifient que l'étiquette d'un type défini au moyen du type ouvert "ObjectClassFieldType" (type champ de classe d'objets) est indéterminée s'il s'agit d'un champ de type, d'un champ de valeur de type variable, ou d'un champ d'ensemble de valeurs de type variable. Ce type est par conséquent défini comme un type ASN.1, et le codage complet est alors identique à celui d'une valeur du type affecté (y compris le champ de l'identificateur).

8.1.3 Champ de longueur

8.1.3.1 Deux formes de champs de longueur sont spécifiées:

- a) la forme définie (voir 8.1.3.3);
- b) la forme indéfinie (voir 8.1.3.6).

8.1.3.2 Un expéditeur utilisera:

- a) la forme définie (voir 8.1.3.3) si le codage est primitif;
- b) au choix la forme définie (voir 8.1.3.3) ou la forme indéfinie (voir 8.1.3.6) si le codage est structuré et immédiatement disponible dans son entier;
- c) la forme indéfinie (voir 8.1.3.6) si le codage est structuré et pas immédiatement disponible dans son entier.

8.1.3.3 Pour la forme définie, le champ de longueur comportera un ou plusieurs octets, et représentera le nombre d'octets du champ de contenu, en utilisant au choix de l'expéditeur la forme courte (voir 8.1.3.4) ou la forme longue (voir 8.1.3.5).

NOTE – La forme courte ne peut être utilisée que si le nombre des octets du champ de contenu est inférieur ou égal à 127.

8.1.3.4 Dans la forme courte, le champ de longueur comporte un seul octet dans lequel le bit 8 a la valeur zéro et les bits 7 à 1 représentent le nombre d'octets du champ de contenu (éventuellement zéro) sous forme d'un entier binaire non signé, le bit 7 ayant le poids le plus fort.

Exemple

L = 38 peut être codé 00100110₂.

8.1.3.5 Dans la forme longue, le champ de longueur comprend un octet initial suivi d'un ou plusieurs octets. L'octet initial est codé comme suit:

- a) le bit 8 est à un;

- b) les bits 7 à 1 représentent le nombre des octets suivants du champ de longueur, sous forme d'un entier binaire non signé, le bit 7 ayant le poids le plus fort;
- c) la valeur 11111111_2 ne sera pas utilisée.

NOTE 1 – Cette restriction est introduite en vue d'une future extension possible.

Les bits 8 à 1 du premier octet suivant, suivis des bits 8 à 1 du deuxième octet suivant, suivis ainsi de suite des bits 8 à 1 de chacun des octets suivants jusques et y compris le dernier octet suivant, représentent sous forme d'un entier binaire non signé le nombre d'octets du champ de contenu, le bit 8 du premier octet suivant ayant le poids le plus fort.

Exemple

L = 201 sera codé:

10000001_2 11001001_2

NOTE 2 – Dans la forme longue, l'expéditeur peut choisir d'utiliser plus d'octets pour le champ de longueur que le minimum nécessaire.

8.1.3.6 Dans la forme indéfinie, le champ de longueur, qui ne comporte alors qu'un seul octet, indique que le champ de contenu est terminé par une séquence de fin de contenu (voir 8.1.5).

8.1.3.6.1 Cet octet aura son bit 8 à un et ses bits 7 à 1 à zéro.

8.1.3.6.2 Si la forme longue est utilisée, la séquence de fin de contenu (voir 8.1.5) figurera dans le codage à la suite du champ de contenu.

8.1.4 Champ de contenu

Le champ de contenu comportera zéro, un ou plusieurs octets et représentera la valeur de données conformément au codage spécifié dans les articles suivants.

NOTE – Le champ de contenu dépend du type de la valeur de données; les articles ci-dessous suivent l'ordre des définitions des types dans l'ASN.1.

8.1.5 Séquence de fin de contenu

La séquence de fin de contenu figurera dans le codage si la longueur est codée conformément au 8.1.3.6; sinon, elle n'y figurera pas.

La séquence de fin de contenu sera constituée de deux octets mis à zéro.

NOTE – La séquence de fin de contenu peut être considérée comme le codage d'une valeur dont l'étiquette est de classe universelle, la forme primitive, le numéro d'étiquette égal à zéro et le contenu absent:

| Identificateur = fin de contenu | Longueur | Contenu |
|------------------------------------|-----------|---------|
| 00_{16} | 00_{16} | Néant |

8.2 Codage d'une valeur booléenne

8.2.1 Le codage d'une valeur booléenne sera de forme primitive. Le champ de contenu comportera un seul octet.

8.2.2 Si la valeur booléenne est égale à "Faux", l'octet prendra la valeur zéro.

Si la valeur booléenne est égale à "Vrai", l'octet prendra n'importe quelle valeur différente de zéro, au choix de l'expéditeur.

Exemple – Si elle est de type BOOLEAN, la valeur "Vrai" peut être codée:

| Identificateur = booléen | Longueur | Contenu |
|-----------------------------|------------------|------------------|
| 01 ₁₆ | 01 ₁₆ | FF ₁₆ |

8.3 Codage d'une valeur entière

8.3.1 Le codage d'une valeur entière est de forme primitive. Le champ de contenu comportera un ou plusieurs octets.

8.3.2 Si le champ de contenu du codage d'un entier comporte plus d'un octet, les bits du premier octet et le bit 8 du deuxième octet:

- a) ne seront pas tous des 1;
- b) ne seront pas tous des 0.

NOTE – Ces règles assurent le codage d'une valeur entière sur le plus petit nombre d'octets.

8.3.3 Le champ de contenu sera la représentation binaire en complément à deux de l'entier, et sera composé des bits 8 à 1 du premier octet, suivis des bits 8 à 1 du deuxième octet, et ainsi de suite jusques et y compris le dernier des octets du champ de contenu.

NOTE – La valeur d'un nombre représenté en notation binaire en complément à deux est obtenue en numérotant les bits des octets du champ de contenu à partir du bit 1 du dernier octet (bit 0) jusqu'au bit 8 du premier octet. A chaque bit est affecté un poids de 2^N correspondant à son rang N. La valeur du nombre est obtenue en faisant la somme des valeurs numériques affectées à chacun des bits mis à un (sauf le bit 8 du premier octet), de laquelle on retranche la valeur affectée au bit 8 du premier octet si celui-ci est à un.

8.4 Codage d'une valeur énumérée

Le codage d'une valeur énumérée est celui de l'entier auquel elle est associée.

NOTE – Il s'agit d'une forme primitive.

8.5 Codage d'une valeur réelle

8.5.1 Le codage d'une valeur réelle est de forme primitive.

8.5.2 Si le réel a la valeur zéro, le codage ne comportera pas de champ de contenu.

8.5.3 Si le réel est différent de zéro, il sera représenté dans une base B' choisie par l'expéditeur. Si B' est égal à 2, 8 ou 16, le codage binaire spécifié au 8.5.5 sera utilisé. Si B' est égal à 10, le codage caractère spécifié au 8.5.6 sera utilisé.

NOTE – La forme utilisée pour l'enregistrement, la génération ou le traitement par les expéditeurs ou les destinataires et la forme utilisée dans la notation de valeur ASN.1, sont toutes indépendantes de la base utilisée en transfert.

8.5.4 Le bit 8 du premier octet du champ de contenu sera codé comme suit:

- a) bit 8 = 1: le codage binaire spécifié au 8.5.5 s'applique;
- b) bit 8 = 0 et bit 7 = 0: le codage décimal spécifié au 8.5.6 s'applique;
- c) bit 8 = 0 et bit 7 = 1: une valeur réelle spéciale "SpecialRealValue" (voir la Rec. UIT-T X.680 | ISO/CEI 8824-1) est codée conformément au 8.5.7.

8.5.5 Lorsqu'un codage binaire est utilisé (bit 8 = 1) et que la mantisse M est différente de zéro, la valeur sera représentée par un signe S, un entier non négatif N et un facteur d'échelle F:

$$M = S \times N \times 2^F$$

avec: $0 \leq F < 4$,

et $S = \pm 1$

NOTE – Le facteur d'échelle binaire F est nécessaire dans certaines circonstances pour aligner le point implicite de la mantisse sur la position imposée par les règles de codage. Cet alignement ne peut pas toujours être assuré par l'exposant E. En effet, si la base B' utilisée dans le codage est égale à 8 ou à 16, le point implicite ne peut être déplacé respectivement que par pas de 3 ou 4 bits par modification de la composante E. Il sera donc parfois nécessaire d'utiliser un facteur d'échelle binaire F différent de 0 pour déplacer le point implicite jusqu'à la position requise.

8.5.5.1 Le bit 7 du premier octet du champ de contenu sera mis à 1 si $S = -1$, et à 0 sinon.

8.5.5.2 Les bits 6 et 5 du premier octet du champ de contenu représenteront la valeur de la base B' codée comme suit:

| <i>Bits 6 et 5</i> | <i>Base</i> |
|--------------------|---|
| 00 | base 2 |
| 01 | base 8 |
| 10 | base 16 |
| 11 | Réservé aux versions ultérieures de la présente Recommandation Norme internationale |

8.5.5.3 Les bits 4 et 3 du premier octet du champ de contenu représentent la valeur du facteur d'échelle F sous la forme d'un entier binaire non signé.

8.5.5.4 Les bits 2 et 1 du premier octet du champ de contenu représentent le format de l'exposant codé comme suit:

- si les bits (2, 1) valent 00, le deuxième octet du champ de contenu contient la valeur de l'exposant en représentation binaire en complément à deux;
- si les bits (2, 1) valent 01, le deuxième et le troisième octet du champ de contenu contiennent la valeur de l'exposant en représentation binaire en complément à deux;
- si les bits (2, 1) valent 10, le deuxième, le troisième et le quatrième octet du champ de contenu contiennent la valeur de l'exposant en représentation binaire en complément à deux;
- si les bits (2, 1) valent 11, le deuxième octet de contenu représente le nombre d'octets, disons X, (sous forme d'un entier binaire non signé) utilisé pour coder la valeur de l'exposant, et les troisièmes à $(X + 3)^{\text{e}}$ (compris) octets de contenu représentent la valeur de l'exposant sous la forme d'un entier binaire en complément à deux, la valeur de X étant au moins égale à un; les neuf premiers bits de l'exposant transmis ne doivent pas être tous à 0 ni tous à 1.

8.5.5.5 Les octets restants du champ de contenu représentent la valeur de l'entier N (voir 8.5.5) sous la forme d'un entier binaire non signé.

NOTES

1 Dans les règles de codage de base (BER) non canoniques, il n'est pas prescrit de normaliser la virgule flottante de la mantisse. Ceci permet au concepteur de transmettre le champ de la mantisse sans avoir à en déplacer préalablement les octets en mémoire. Quant aux règles de codage canoniques et aux règles de codage distinctives, la position de la virgule flottante y est normalisée, et il faut déplacer itérativement les octets de la mantisse (à moins que celle-ci soit nulle) jusqu'à ce que le bit le moins significatif soit égal à 1.

2 Cette représentation des nombres réels est très différente des formats normalement utilisés dans les calculateurs à virgule flottante, mais elle est conçue pour faciliter la conversion vers ou depuis de tels formats (voir Annexe C).

8.5.6 Quand le codage décimal est utilisé (bits 8 et 7 = 00), tous les octets du champ de contenu venant à la suite du premier octet de ce champ constituent un champ, au sens de ISO 6093, de longueur choisie par l'expéditeur et codé conformément à cette norme. Le type de représentation ISO 6093 utilisé est indiqué par les bits 6 à 1 du premier octet du champ de contenu:

| <i>Bits 6 à 1</i> | <i>Représentation des nombres</i> |
|-------------------|-----------------------------------|
| 00 0001 | Forme NR1 ISO 6093 |
| 00 0010 | Forme NR2 ISO 6093 |
| 00 0011 | Forme NR3 ISO 6093 |

Les valeurs restantes des bits 6 à 1 sont réservées à l'usage des versions ultérieures de la présente Recommandation | Norme internationale.

La documentation associée ne doit pas spécifier de facteur d'échelle (voir ISO 6093).

NOTES

1 Comme ISO 6093, la présente Recommandation | Norme internationale préconise de garder au moins un chiffre à gauche de la virgule décimale sans que ceci soit obligatoire.

2 L'utilisation de la forme normalisée (voir ISO 6093) est au choix de l'expéditeur et n'a pas de signification particulière.

8.5.7 S'il faut coder une valeur réelle spéciale "SpecialRealValues" (bits 8 à 7 = 01), le champ de contenu comportera un seul octet pouvant prendre les valeurs suivantes:

01000000 la valeur est PLUS-INFINITY ($+\infty$)

01000001 la valeur est MINUS-INFINITY ($-\infty$)

Toutes les autres valeurs avec les bits 8 et 7 égaux à 01 sont réservées aux versions ultérieures de la présente Recommandation | Norme internationale.

8.6 Codage d'une valeur de type chaîne binaire

8.6.1 La représentation d'une valeur de type chaîne binaire est un codage primitif ou structuré, au choix de l'expéditeur.

NOTE – S'il faut transférer une partie d'une chaîne binaire avant que la chaîne complète ne soit disponible, il faut utiliser le codage structuré.

8.6.2 Le champ de contenu du codage primitif contient un octet initial suivi de zéro, un ou plusieurs octets.

8.6.2.1 La chaîne binaire, depuis le premier bit jusqu'au dernier, doit être placée dans les bits 8 à 1 du premier octet suivant, suivis des bits 8 à 1 de l'octet d'après, puis des bits 8 à 1 de chacun des octets suivants, suivis d'autant de bits que nécessaire dans le dernier octet, en commençant par le bit 8.

NOTE – Les termes "premier bit" et "dernier bit" sont définis dans la Rec. UIT-T X.680 | ISO/CEI 8824-1.

8.6.2.2 Le premier octet indiquera le nombre de bits non utilisés dans l'octet final sous la forme d'un entier binaire avec le bit de plus faible poids en position 1. Ce nombre appartient à l'intervalle [0..7].

8.6.2.3 Si la chaîne binaire est vide, l'octet initial sera mis à 0 et ne sera suivi d'aucun octet.

8.6.2.4 Lorsque les dispositions du paragraphe 19.7 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 s'appliquent, un codeur/décodeur conforme aux règles BER peut ajouter ou supprimer des bits de fin à 0 de la valeur transmise.

NOTE – Si une valeur de chaîne binaire ne comporte pas de bits à 1, le codeur peut (au choix de l'expéditeur) coder la valeur sur une longueur nulle sans champ de contenu, ou la coder sous la forme d'une chaîne binaire comportant un ou plusieurs bits nuls.

8.6.3 Le champ de contenu du codage structuré comporte zéro, une ou plusieurs valeurs codées encapsulées.

NOTE – Chacune de ces valeurs codées comprend des champs d'identification, de longueur, de contenu, et éventuellement de fin de contenu.

8.6.4 Pour coder une valeur de chaîne binaire de cette façon, il faut la segmenter. Chaque segment est constitué d'une série de bits consécutifs de la valeur et contient (sauf éventuellement le dernier segment) un nombre de bits multiple de huit. Chaque bit de la valeur globale se trouve dans un et un seul segment, mais aucune signification ne doit être accordée aux limites des segments.

NOTE – Un segment peut être de taille nulle, c'est-à-dire qu'il peut ne contenir aucun octet.

8.6.4.1 Dans le champ de contenu, chaque codage représente un segment de la chaîne binaire totale, le codage résultant d'une application récursive du présent point. Dans cette application récursive, chaque segment est traité comme s'il s'agissait d'une chaîne binaire. Les segments codés figureront dans le champ de contenu dans l'ordre dans lequel leurs bits apparaissent dans la valeur globale.

NOTES

1 Par suite de cette récursivité, chaque codage de champ de contenu peut être lui-même de type primitif ou structuré. Toutefois, les codages de ce genre seront généralement de type primitif.

2 En particulier, les étiquettes de champ de contenu seront toujours de classe universelle, numéro 3.

8.6.4.2 Exemple – Si elle est du type chaîne binaire, la valeur '0A3B5F291CD'H peut être codée comme suit sous forme primitive:

| Identificateur = chaîne binaire | Longueur | Contenu |
|------------------------------------|------------------|------------------------------|
| 03 ₁₆ | 07 ₁₆ | 040A3B5F291CD0 ₁₆ |

La valeur ci-dessus peut également être codée comme suit sous une forme structurée:

| Identificateur = chaîne binaire | Longueur | Contenu | | |
|--|------------------------------|--------------------------------------|--------------------------------------|--|
| 23 ₁₆ | 80 ₁₆ | Identificateur = chaîne binaire | Longueur | Contenu |
| EOC = fin de chaîne 00 ₁₆ | Longueur 00 ₁₆ | 03 ₁₆ 03 ₁₆ | 03 ₁₆ 05 ₁₆ | 000A3B ₁₆ 045F291CD0 ₁₆ |

8.7 Codage d'une valeur de type chaîne d'octets

8.7.1 Le codage d'une chaîne d'octets pourra être de forme primitive ou structurée au choix de l'expéditeur.

NOTE – Lorsqu'il est nécessaire de transférer une partie d'une chaîne d'octets avant que la chaîne complète ne soit disponible, il faut utiliser le codage structuré.

8.7.2 Le champ de contenu d'un codage primitif contient zéro, un ou plusieurs octets égaux un à un aux octets de la valeur de données, et placés dans le même ordre que ceux-ci, le bit le plus significatif de chaque octet du champ de contenu étant aligné avec le bit le plus significatif de l'octet correspondant de la valeur de données.

8.7.3 Le champ de contenu d'un codage structuré comportera zéro, un ou plusieurs codages.

NOTE – Chacun de ces codages comportera un champ d'identification, un champ de longueur, un champ de contenu et, s'il est de forme structurée, un champ de fin de contenu.

8.7.3.1 Pour coder une chaîne d'octets de cette façon, il faut la segmenter. Chaque segment est constitué d'une suite d'octets consécutifs de la valeur. Les limites entre segments n'ont aucune signification particulière.

NOTE – Un segment peut être de taille nulle, c'est-à-dire ne contenir aucun octet.

8.7.3.2 Chaque codage du champ de contenu représente un segment de la chaîne d'octets totale, le codage de chacun de ces segments résultant d'une application récursive du présent paragraphe. Dans cette application récursive, chaque segment est traité comme une valeur de chaînes d'octets. Les codages des segments figureront dans le champ de contenu dans l'ordre dans lequel leurs octets apparaissent dans la valeur totale.

NOTES

1 Par suite de cette récursivité, chaque codage du champ de contenu peut être lui-même de forme primitive ou structurée. Toutefois, les codages de ce genre seront généralement de forme primitive.

2 En particulier, les étiquettes du champ de contenu sont toujours de classe universelle, numéro 4.

8.8 Codage d'une valeur vide

8.8.1 Le codage d'une valeur vide est de forme primitive.

8.8.2 Le champ de contenu ne comportera aucun octet.

NOTE – Le champ de longueur contient la valeur zéro.

Exemple – Si elle est de type vide, la valeur NÉANT peut être codée comme suit:

| | |
|------------------------------|------------------|
| <i>Identificateur = vide</i> | <i>Longueur</i> |
| 05 ₁₆ | 00 ₁₆ |

8.9 Codage d'une valeur de type séquence

8.9.1 Le codage d'une séquence sera de forme structurée.

8.9.2 Le champ de contenu sera constitué du codage complet d'une valeur de données pour chacun des types énumérés dans la définition ASN.1 du type séquence, dans l'ordre de leur apparition dans la définition, sauf si le type a été qualifié par le mot clé "OPTIONAL" (optionnel) ou par le mot clé "DEFAULT" (valeur par défaut).

8.9.3 Sans que cela soit nécessaire, le codage d'une valeur de données peut figurer pour un type qui a été qualifié par le mot clé "OPTIONAL" ou par le mot clé "DEFAULT". S'il figure, il doit apparaître dans le codage au point correspondant à l'apparition de ce type dans la définition ASN.1.

Exemple – Si elle est du type

SEQUENCE {nom IA5String, ok BOOLEAN}

la valeur

{nom "Smith", ok Vrai}

peut être codée comme suit:

| | | | | |
|------------------|------------------|------------------|------------------|------------------|
| Identificateur | | | | |
| = séquence | Longueur | Contenu | | |
| 30 ₁₆ | 0A ₁₆ | | | |
| | | Chaîne IA5 | Longueur | Contenu |
| | | 16 ₁₆ | 05 ₁₆ | "Smith" |
| | | Booléen | Longueur | Contenu |
| | | 01 ₁₆ | 01 ₁₆ | FF ₁₆ |

8.10 Codage d'une valeur de type séquence-de

8.10.1 Le codage d'une séquence-de sera de forme structurée.

8.10.2 Le champ de contenu comportera zéro, un ou plusieurs codages complets de valeurs de données du type indiqué dans la définition ASN.1.

8.10.3 L'ordre des codages des valeurs de données sera identique à celui des valeurs de données de la valeur séquence-de à coder.

8.11 Codage d'une valeur de type ensemble

8.11.1 Le codage d'un ensemble sera de forme structurée.

8.11.2 Le champ de contenu comportera le codage complet d'une valeur de données pour chacun des types énumérés dans la définition ASN.1 du type d'ensemble, dans un ordre choisi par l'expéditeur, sauf si le type a été qualifié par le mot clé "OPTIONAL" (optionnel) ou "DEFAULT" (valeur par défaut).

8.11.3 Sans que cela soit nécessaire, le codage d'une valeur de données peut figurer pour un type qui a été qualifié par le mot clé "OPTIONAL" ou "DEFAULT".

NOTE – L'ordre des valeurs de données d'une valeur d'ensemble n'est pas significatif et n'impose aucune contrainte quant à leur ordre de transfert.

8.12 Codage d'une valeur de type ensemble-de

8.12.1 Le codage d'un ensemble-de sera de forme structurée.

8.12.2 Le texte du 8.10.2 s'applique.

8.12.3 L'ordre des valeurs de données ne doit pas nécessairement être conservé par le codage et le décodage qui le suit.

8.13 Codage d'une valeur de type choix

Le codage d'une valeur de type choix est le même que celui d'une valeur du type choisi.

NOTES

- 1 Le codage peut être de forme primitive ou structurée selon le type choisi.
- 2 L'étiquette utilisée dans le champ d'identification est celle du type choisi, telle qu'elle est spécifiée dans la définition ASN.1 du type choix.

8.14 Codage d'une valeur étiquetée

8.14.1 Le codage d'une valeur étiquetée dérivera du codage complet de la valeur de données correspondante du type apparaissant dans la notation du type étiqueté "TaggedType" (appelé codage de base) conformément aux spécifications des 8.14.2 et 8.14.3.

8.14.2 Si l'étiquetage implicite (voir la Rec. UIT-T X.680 | ISO/CEI 8824-1, paragraphe 28.6) n'a pas été utilisé dans la définition du type, le codage sera de forme structurée et le champ de contenu contiendra le codage de base complet.

8.14.3 Si l'étiquetage implicite a été utilisé dans la définition du type:

- a) le codage sera de forme structurée si le codage de base est de forme structurée, et de forme primitive dans le cas contraire;
- b) le champ de contenu sera identique au champ de contenu du codage de base.

Exemple – Avec les définitions des types ASN.1 suivants (dans un environnement à étiquetage explicite):

Type1 ::= VisibleString -- chaîne visible

Type2 ::= [APPLICATION 3] IMPLICIT Type1

Type3 ::= [2] Type2

Type4 ::= [APPLICATION 7] IMPLICIT Type3

Type5 ::= [2] IMPLICIT Type2

la valeur

"Jones"

sera codée comme suit:

Pour le Type1:

| VisibleString | Longueur | Contenu |
|------------------|------------------|--------------------------|
| 1A ₁₆ | 05 ₁₆ | 4A6F6E6573 ₁₆ |

Pour le Type2:

| [Application 3] | Longueur | Contenu |
|------------------|------------------|--------------------------|
| 43 ₁₆ | 05 ₁₆ | 4A6F6E6573 ₁₆ |

Pour le Type3:

| [2] | Longueur | Contenu |
|------------------|------------------|--------------------------|
| A2 ₁₆ | 07 ₁₆ | [Application 3] |
| | | 43 ₁₆ |
| | | [Application 3] |
| | | Longueur |
| | | 05 ₁₆ |
| | | Contenu |
| | | 4A6F6E6573 ₁₆ |

Pour le Type4:

| [Application 7] | Longueur | Contenu |
|------------------|------------------|--------------------------|
| 67 ₁₆ | 07 ₁₆ | [Application 3] |
| | | 43 ₁₆ |
| | | [Application 3] |
| | | Longueur |
| | | 05 ₁₆ |
| | | Contenu |
| | | 4A6F6E6573 ₁₆ |

Pour le Type5:

| [2] | Longueur | Contenu |
|------------------|------------------|--------------------------|
| 82 ₁₆ | 05 ₁₆ | 4A6F6E6573 ₁₆ |

8.15 Codage d'une valeur de type ouvert

Une valeur de type ouvert est aussi une valeur d'un quelconque autre type ASN.1. Le codage d'une telle valeur sera le codage complet de la valeur considérée comme appartenant à l'autre type.

8.16 Codage d'une valeur de type instance-de

8.16.1 Le codage d'une valeur du type instance-de est le codage de base du type de séquence suivant avec la valeur spécifiée au 8.16.2:

```
[UNIVERSAL 8] IMPLICIT SEQUENCE
{
    type-id    <DefinedObjectClass>.&id,
    value      [0] EXPLICIT <DefinedObjectClass>.&Type
}
```

où "<DefinedObjectClass>" est remplacé par la classe "DefinedObjectClass" particulière utilisée dans la notation d'instance-de "InstanceOfType".

NOTE – Lorsque la valeur appartient à un seul type ASN.1, et qu'elle est codée à l'aide des règles de codage de base, son codage est alors identique à la valeur correspondante du type externe, dans lequel la variante "syntax" est utilisée pour représenter la valeur abstraite.

8.16.2 La valeur des composantes du type séquence au 8.16.1 sera la même que les valeurs des composantes correspondantes du type associé du C.7 de la Rec. UIT-T X.681 | ISO/CEI 8824-2.

8.17 Codage d'une valeur de type valeur de données de présentation encapsulée

8.17.1 Il existe deux sous-règles pour coder les valeurs de données de présentation (pdv) encapsulées: la règle EP-A d'affectation d'index, et la règle EP-B d'utilisation d'index. Pour chaque valeur d'identificateur dans la valeur abstraite, la première occurrence (dans le codage de la valeur de présentation entière) d'une valeur de données de présentation encapsulée ayant cette valeur d'identificateur sera codée selon la règle EP-A.

8.17.2 Sous réserve des restrictions énumérées ci-dessous, les valeurs ultérieures ayant la même valeur d'identificateur seront codées selon la règle EP-B, l'index recevant la même valeur que celui du codage EP-A correspondant.

8.17.3 Les conditions d'utilisation de la sous-règle EP-B sont énumérées ci-dessous. Si une de ces conditions n'est pas satisfaite, il faut utiliser la sous-règle EP-A à la place:

- l'index appartient à l'intervalle [0 .. 255];
- le codage de cette instance de valeur de données de présentation encapsulée comporte un nombre de bits multiple de huit;
- la taille du codage n'est pas supérieure à la longueur maximale pouvant être identifiée par un code de longueur en forme longue.

NOTE – Cette dernière condition n'aura vraisemblablement pas de caractère restrictif en pratique.

8.17.4 A la première occurrence d'un codage selon la règle EP-A, l'index aura la valeur 0, puis il sera incrémenté de 1 à chaque occurrence ultérieure.

8.17.5 EXPLICATION – Ainsi, pour chaque valeur d'identificateur, il y aura un codage EP-A (relativement inefficace) comportant à la fois une valeur d'index unique et la valeur complète d'identification, suivi d'un nombre arbitraire de codages EP-B (efficaces) rattachés au codage EP-A par la valeur d'index. Comme le codage EP-B utilise un seul octet pour l'index, et un comptage en octets pour le codage, il ne peut être utilisé si la valeur d'index excède 255 (256 identificateurs différents utilisés), ou si le codage ne comporte pas un nombre de bits multiple de 8. Dans ces deux derniers cas, le codage EP-A sera utilisé pour toutes les occurrences.

8.17.6 Le codage EP-A sera le codage de base du type de séquence suivant, après application de l'étiquetage automatique AUTOMATIC TAGS conformément aux 22.6 et 26.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1:

```
[UNIVERSAL 11] IMPLICIT SEQUENCE {
    index                INTEGER,
    identification       CHOICE {
        syntaxes         SEQUENCE {
            abstract     OBJECT IDENTIFIER,
            transfer      OBJECT IDENTIFIER },
        syntax           OBJECT IDENTIFIER,
        presentation-context-id INTEGER,
        context-negotiation SEQUENCE {
            presentation-context-id INTEGER,
            transfer-syntax OBJECT IDENTIFIER },
        transfer-syntax  OBJECT IDENTIFIER,
        fixed            NULL },
    data-value          BIT STRING }
```

8.17.7 La valeur "data-value" représentera une valeur de donnée abstraite utilisant la syntaxe de transfert identifiée, la valeur "index" sera déterminée comme ci-dessus, et la valeur des autres champs sera la même que celles des valeurs apparaissant dans la valeur abstraite.

NOTES

1 La composante "index" n'est pas définie dans la syntaxe abstraite car il est prévu qu'elle soit fournie uniquement au niveau des règles de codage (de la même manière que le champ fin de contenu s'applique au niveau des règles de codage).

2 Les deux formes de "data-value" en syntaxe abstraite sont codées de manière identique en syntaxe de transfert sous la forme d'une chaîne binaire.

8.17.8 Le codage EP-B sera un codage BER du type ASN.1 suivant:

[UNIVERSAL 11] IMPLICIT OCTET STRING

où

- a) le codage est primitif;
- b) le champ de longueur est codé selon la forme définie courte ou longue, au choix de l'expéditeur;
- c) le champ de contenu comprend dans son premier octet, l'index, interprété comme un entier compris entre 0 et 255, suivi par le code correspondant à la valeur de données.

NOTE – Le destinataire distinguera un codage EP-A d'un codage EP-B par la valeur du bit primitif/structuré.

8.18 Codage d'une valeur de type externe

8.18.1 Le codage d'une valeur de type externe sera le codage de base du type de séquence suivant, supposé défini dans un environnement d'étiquetage explicite EXPLICIT TAGS, avec une valeur selon les spécifications des points suivants:

[UNIVERSAL 8] IMPLICIT SEQUENCE {

| | |
|------------------------------|---------------------------------------|
| direct-reference | OBJECT IDENTIFIER OPTIONAL, |
| indirect-reference | INTEGER OPTIONAL, |
| data-value-descriptor | ObjectDescriptor OPTIONAL, |
| encoding | CHOICE { |
| single-ASN1-type | [0] ABSTRACT-SYNTAX.&Type, |
| octet-aligned | [1] IMPLICIT OCTET STRING, |
| arbitrary | [2] IMPLICIT BIT STRING } } |

NOTE – Ce type de séquence est le même que celui qui est spécifié dans la Rec. X.208 du CCITT (1988) | ISO/CEI 8824 (1990), et le codage résultant d'une valeur de type externe est inchangé par rapport à ces spécifications.

8.18.2 La valeur des champs dépend de la valeur abstraite à transmettre, qui est une valeur du type spécifié au 30.5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

8.18.3 Le descripteur de valeur de données "data-value-descriptor" ci-dessus sera présent si et seulement si ce descripteur est présent dans la valeur abstraite, et aura dans ce cas la même valeur.

8.18.4 Le Tableau 2, qui énumère les différentes variantes d'identification de la valeur abstraite (voir 30.5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1), indique les conditions de présence ou d'absence des champs "direct-reference" (référence directe) et "indirect-reference" (référence indirecte), et s'ils sont présents, la valeur prise par le champ abstrait.

Tableau 2 – Différentes variantes du codage du champ d'identification identifier

| identification | direct-reference (référence directe) Rec. UIT-T X.690 (1994 F) | indirect-reference (référence indirecte) Remplacée par une version plus récente |
|----------------|--|---|
| syntaxes | *** NE PEUT SE PRODUIRE *** | *** NE PEUT SE PRODUIRE *** |

Remplacée par une version plus récente ISO/CEI 8825-1 : 1995 (F)

8.18.5 La valeur de données sera codée conformément à la syntaxe de transfert identifiée par le codage, et sera placée dans un des champs possibles du choix de codage "encoding" selon les spécifications ci-dessous.

8.18.6 Si la valeur de données appartient à un seul type de données ASN.1, et si les règles de codage pour cette valeur de données sont les mêmes que pour le type externe de données EXTERNAL, alors l'application expéditrice utilisera l'un quelconque des codages suivants au choix:

| | |
|------------------|------------------------|
| single-ASN1-type | (type ASN.1 simple) |
| octet-aligned | (alignement sur octet) |
| arbitrary | (arbitraire) |

8.18.7 Si le codage de la valeur de données, effectué selon le codage agréé ou négocié, comporte un nombre entier d'octets, alors l'application expéditrice utilisera l'un quelconque des codage suivants au choix:

| | |
|---------------|------------------------|
| octet-aligned | (alignement sur octet) |
| arbitrary | (arbitraire) |

NOTE – Si une valeur de données est une suite de types ASN.1, et que la syntaxe de transfert spécifie une simple concaténation des chaînes d'octets générées par l'application des règles de codage de base de l'ASN.1 à chacun de ces types, alors la valeur de données relève de la présente catégorie et non pas du 8.18.6.

8.18.8 Si le codage de la valeur de données, effectué selon le codage agréé ou négocié, ne comporte pas un nombre entier d'octets, alors le choix de codage suivant sera utilisé:

| | |
|-----------|--------------|
| arbitrary | (arbitraire) |
|-----------|--------------|

8.18.9 Si le codage adopté est le "single-ASN1-type" (type ASN.1 simple), le type ASN.1 remplacera alors le type ouvert, avec une valeur égale à la valeur de données à coder.

NOTE – Le domaine possible des valeurs pouvant être prises par un type ouvert est déterminé par la valeur d'identificateur d'objet enregistrée associée à la référence directe "direct-reference", et/ou par la valeur entière associée à la référence indirecte "indirect-reference".

8.18.10 Si le codage choisi est l'alignement sur octet "octet-aligned", la valeur de données sera codée selon la syntaxe de transfert agréée ou négociée, et le résultat formera la valeur de la chaîne d'octets.

8.18.11 Si le codage choisi est le codage arbitraire "arbitrary", la valeur de données sera codée selon la syntaxe de transfert agréée ou négociée, et le résultat formera la valeur de la chaîne binaire.

8.19 Codage d'une valeur d'identificateur d'objet

8.19.1 Le codage d'une valeur d'identificateur d'objet sera de forme primitive.

8.19.2 Le champ de contenu sera constitué d'une liste (ordonnée) des codages concaténés des sous-identificateurs (voir 8.19.3 et 8.19.4).

Chaque sous-identificateur est représenté par une suite d'un ou plusieurs octets. Le bit 8 de chaque octet indique s'il est le dernier de la suite: le bit 8 du dernier octet est mis à 0, le bit 8 de chaque octet précédent est mis à 1. Les bits 7 à 1 des octets de la suite codent collectivement le sous-identificateur. Conceptuellement, ces groupes de bits sont concaténés sous la forme d'un entier non signé dont le bit de plus fort poids est le bit 7 du premier octet et dont le bit de plus faible poids est le bit 1 du dernier octet. Le sous-identificateur sera codé sur le plus petit nombre d'octets possible, c'est-à-dire que l'octet de début ne devra pas avoir la valeur 80_{16} .

8.19.3 Le nombre (N) de sous-identificateurs sera inférieur de 1 au nombre de composants d'identificateur d'objet de la valeur d'identificateur d'objet à coder.

8.19.4 La valeur numérique du premier sous-identificateur est obtenue à partir des valeurs des **deux** premiers composants d'identificateur d'objet de la valeur d'identificateur d'objet à coder, en appliquant la formule:

$$(X*40) + Y$$

où X et Y sont respectivement les valeurs du premier et du deuxième composant de l'identificateur d'objet.

NOTE – Ce regroupement des deux premiers composants de l'identificateur d'objet suppose que trois valeurs seulement sont affectées à partir d'une racine, et au plus 39 valeurs aux valeurs suivantes à partir des nœuds atteints par $X = 0$ et $X = 1$.

8.19.5 La valeur numérique du $i^{\text{ème}}$ sous-identificateur ($2 \leq i \leq N$) est celle du $(i + 1)^{\text{ème}}$ composant de l'identificateur d'objet.

Exemple – Une valeur d'identificateur d'objet OBJECT IDENTIFIER de:

{joint-iso-ccitt 100 3}

qui est la même que:

{2 100 3}

a un premier sous-identificateur de 180 et un second sous-identificateur de 3. Le codage résultant est:

| Identificateur d'objet | Longueur | Contenu |
|------------------------|------------------|----------------------|
| 06 ₁₆ | 03 ₁₆ | 813403 ₁₆ |

8.20 Codage d'une valeur de type chaîne de caractères avec restriction

8.20.1 La valeur de données est constituée d'une chaîne de caractères du jeu de caractères spécifié dans la définition de type ASN.1.

8.20.2 Chaque valeur de données est codée indépendamment des autres valeurs de données de même type.

8.20.3 Chaque type de chaîne de caractères est codé comme s'il avait été déclaré

[UNIVERSAL x] IMPLICIT OCTET STRING

où x est le numéro de l'étiquette de classe universelle affecté au type de la chaîne de caractères dans la Rec. UIT-T X.680 | ISO/CEI 8824-1. La valeur de la chaîne d'octets est spécifiée aux 8.20.4 et 8.20.5.

8.20.4 Lorsqu'un type chaîne de caractères est spécifié dans la Rec. UIT-T X.680 | ISO/CEI 8824-1 par référence directe à un tableau énumératif ("NumericString" chaîne numérique et "PrintableString" chaîne imprimable), la valeur de la chaîne d'octets est celle spécifiée au 8.20.5 pour un type "VisibleString" (chaîne visible) de même valeur que la chaîne de caractères.

8.20.5 Pour les chaînes de caractères avec restriction autres que la chaîne universelle "UniversalString" et la chaîne de table multilingue "BMPString", la chaîne d'octets contiendra les octets spécifiés par ISO 2022 pour le codage en environnement 8 bits, et utilisera la séquence d'échappement et les codes caractères enregistrés conformément à ISO 2375.

8.20.5.1 Seules peuvent être utilisées les séquences d'échappement spécifiées par un des numéros d'enregistrement utilisés pour définir le type chaîne de caractères dans la Rec. UIT-T X.680 | ISO/CEI 8824-1.

8.20.5.2 Au début de chaque chaîne, certains numéros d'enregistrement seront supposés être désignés par G0 et/ou C0 et/ou C1 et invoqués (selon la terminologie de ISO 2022). Le Tableau 3 spécifie ces numéros d'enregistrement pour chaque type, avec la séquence d'échappement supposée qu'ils impliquent.

8.20.5.3 Certains types de chaînes de caractères ne doivent pas contenir des séquences d'échappement explicites dans leurs codages; dans tous les autres cas, toute séquence d'échappement autorisée par 8.20.5.1 peut apparaître à tout moment, y compris en début de codage. Le Tableau 3 énumère les types pour lesquels les séquences d'échappement explicites sont autorisées.

8.20.5.4 L'utilisation des annonceurs est interdite, sauf autorisation explicite par l'utilisateur ASN.1.

NOTE – Le choix d'un type ASN.1 fournit une forme limitée de la fonction d'annonceur. Des protocoles d'application spécifiques peuvent véhiculer des annonceurs dans d'autres éléments de protocole, ou spécifier en détail la façon d'utiliser les annonceurs.

Tableau 3 – Utilisation des séquences d'échappement

| Type | G0 implicite (numéro d'enregistrement) | C0 & C1 implicites (numéro d'enregistrement) | Séquence(s) d'échappement implicite(s) et verrouillage LS0 (s'il y a lieu) | Séquences d'échappement explicites autorisées? |
|--|--|--|--|--|
| Chaîne numérique (NumericString) | 6 | Aucun | ESC 2/8 4/2 LS0 | Non |
| Chaîne imprimable (PrintableString) | 6 | Aucun | ESC 2/8 4/2 LS0 | Non |
| Chaîne télétexte (chaîne T61) [TeletexString (T61String)] | 102 | 106 (C0) 107 (C1) | ESC 2/8 7/5 LS0 ESC 2/1 4/5 ESC 2/2 4/8 | Oui |
| Chaîne vidéotexte (VideotexString) | 102 | 1 (C0) 73 (C1) | ESC 2/8 7/5 LS0 ESC 2/1 4/0 ESC 2/2 4/1 | Oui |
| Chaîne visible (chaîne ISO 646) [VisibleString (ISO646String)] | 6 | Aucun | ESC 2/8 4/2 LS0 | Non |
| Chaîne IA5 (IA5String) | 6 | 1 (C0) | ESC 2/8 4/2 LS0 ESC 2/1 4/0 | Non |
| Chaîne graphique (GraphicString) | 6 | Aucun | ESC 2/8 4/2 LS0 | Oui |
| Chaîne générale (GeneralString) | 6 | 1 (C0) | ESC 2/8 4/2 LS0 ESC 2/1 4/0 | Oui |

NOTE – Beaucoup de caractères d'usage courant (A à Z par exemple) apparaissent dans divers répertoires de caractères avec différents numéros d'enregistrement et séquences d'échappement. Quand les types ASN.1 autorisent les séquences d'échappement, une chaîne de caractères donnée pourra être codée de diverses manières (voir également 7.3).

Exemple – Avec la définition du type ASN.1 suivante:

Name ::= VisibleString

une valeur

"Jones"

peut être codée (sous forme primitive) de la manière suivante:

| | | |
|------------------|------------------|--------------------------|
| Identificateur | | |
| = chaîne visible | Longueur | Contenu |
| 1A ₁₆ | 05 ₁₆ | 4A6F6E6573 ₁₆ |

ou (sous forme structurée de longueur définie) de la manière suivante:

| | | | | |
|------------------|------------------|------------------|------------------|----------------------|
| Identificateur | | | | |
| = chaîne visible | Longueur | Contenu | | |
| 3A ₁₆ | 09 ₁₆ | | | |
| | | Chaîne d'octets | Longueur | Contenu |
| | | 04 ₁₆ | 03 ₁₆ | 4A6F6E ₁₆ |
| | | Chaîne d'octets | Longueur | Contenu |
| | | 04 ₁₆ | 02 ₁₆ | 6573 ₁₆ |

ou (sous forme structurée de longueur indéfinie) de la manière suivante:

| | | | | |
|------------------|------------------|------------------|------------------|----------------------|
| Identificateur | | | | |
| = chaîne visible | Longueur | Contenu | | |
| 3A ₁₆ | 80 ₁₆ | | | |
| | | Chaîne d'octets | Longueur | Contenu |
| | | 04 ₁₆ | 03 ₁₆ | 4A6F6E ₁₆ |

| Chaîne d'octets | Longueur | Contenu |
|------------------|------------------|--------------------|
| 04 ₁₆ | 02 ₁₆ | 6573 ₁₆ |
| Fin de contenu | Longueur | |
| 00 ₁₆ | 00 ₁₆ | |

8.20.6 Les exemples ci-dessus illustrent trois des (nombreuses) formes possibles de codage au choix de l'expéditeur. Le destinataire devra pouvoir traiter toutes les formes permises de codage (voir 7.3).

8.20.7 Dans le type de chaîne universelle "UniversalString", la chaîne d'octets devra contenir les octets spécifiés dans ISO/CEI 10646-1, en utilisant la forme canonique à 4 octets (voir 14.2 de ISO/CEI 10646-1). Les fonctions de contrôle et les signatures ne seront pas utilisées.

8.20.8 Dans le type de chaîne de table multilingue "BMPString", la chaîne d'octets devra contenir les octets spécifiés dans ISO/CEI 10646-1, en utilisant la forme BMP à 2 octets (voir 14.1 de ISO/CEI 10646-1). Les fonctions de contrôle et les signatures ne seront pas utilisées.

8.21 Codage d'une valeur de type chaîne de caractères sans restriction

8.21.1 Deux sous-règles sont utilisées pour coder le type chaîne de caractères sans restriction: la règle CH-A d'affectation d'index, et la règle CH-B d'utilisation d'index. Pour chaque valeur d'identificateur "identification" dans la valeur abstraite, la première occurrence (dans le codage de la valeur de présentation entière) d'une valeur du type chaîne de caractères sans restriction ayant cette valeur d'identificateur sera codée selon la règle CH-A.

8.21.2 Sous réserve des restrictions énumérées ci-dessous, les valeurs ultérieures ayant la même valeur d'identificateur seront codées selon la règle CH-B, l'index recevant la même valeur que celui du codage CH-A correspondant.

8.21.3 Les conditions d'utilisation de la sous-règle CH-B sont énumérées ci-dessous. Si une de ces conditions n'est pas satisfaite, on utilisera à la place la sous-règle CH-A:

- l'index appartient à l'intervalle [0 .. 255];
- la taille du codage n'est pas supérieure à la longueur maximale pouvant être identifiée par un code de longueur en forme longue.

NOTE – Cette dernière condition n'aura vraisemblablement pas de caractère restrictif en pratique.

8.21.4 A la première occurrence d'un codage selon la règle CH-A, l'index aura la valeur 0, puis il sera incrémenté de 1 à chaque occurrence ultérieure.

8.21.5 EXPLICATION – Ainsi, pour chaque valeur d'identificateur "identification", il y aura un codage CH-A (relativement inefficace) comportant à la fois une valeur d'index unique et la valeur complète d'identification, suivi d'un nombre arbitraire de codages CH-B (efficaces) rattachés au codage CH-A par la valeur "index". Comme le codage CH-B utilise un seul octet pour l'index, et un comptage en octets pour le codage, il ne peut être utilisé si la valeur d'index excède 255 (256 identificateurs différents utilisés). Dans ce cas, le codage CH-A sera utilisé pour toutes les occurrences.

8.21.6 Le codage CH-A sera le codage de base du type de séquence suivant, après application de l'étiquetage automatique AUTOMATIC TAGS conformément aux 22.6 et 26.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1:

```
[UNIVERSAL 29] IMPLICIT SEQUENCE {
  index                INTEGER,
  identification        CHOICE {
    syntaxes            SEQUENCE {
      abstract          OBJECT IDENTIFIER,
      transfer          OBJECT IDENTIFIER },
    syntax              OBJECT IDENTIFIER,
    presentation-context-id INTEGER,
    context-negotiation SEQUENCE {
      presentation-context-id INTEGER,
      transfer-syntax    OBJECT IDENTIFIER },
    transfer-syntax    OBJECT IDENTIFIER,
    fixed              NULL },
  string-value         OCTET STRING }
```

8.21.7 La valeur "string-value" représentera la valeur de chaîne de caractères abstraite utilisant la syntaxe de transfert de caractères identifiée, la valeur "index" sera déterminée comme ci-dessus, et la valeur des autres champs sera identique aux valeurs apparaissant dans la valeur abstraite.

NOTES

1 La composante "index" n'est pas définie dans la syntaxe abstraite car il est prévu qu'elle soit fournie uniquement au niveau des règles de codage (de la même manière que le champ fin de contenu s'applique au niveau des règles de codage).

2 Les deux formes de "string-value" en syntaxe abstraite sont codées de manière identique en syntaxe de transfert sous la forme d'une chaîne binaire.

8.21.8 Le codage CH-B sera un codage BER du type ASN.1 suivant:

[UNIVERSAL 29] IMPLICIT OCTET STRING

où

- a) le codage est primitif;
- b) le champ de longueur est codé selon la forme définie courte ou longue, au choix de l'expéditeur;
- c) le champ de contenu comprend dans son premier octet l'index, interprété comme un entier compris entre 0 et 255, suivi par le code correspondant à la valeur de type chaîne.

NOTE – Le destinataire distinguera un codage CH-A d'un codage CH-B selon la valeur du bit primitif/structuré.

9 Règles de codage canoniques

Le codage canonique d'une valeur de données correspond au codage de cette valeur selon les règles du codage de base décrites à l'article 8, avec les restrictions suivantes ainsi que les restrictions énumérées à l'article 11.

9.1 Formes de longueur

S'il est structuré, le codage emploie la forme de longueur indéfinie; s'il est primitif, il comportera un champ de longueur ayant le plus petit nombre d'octets possible (et non comme au 8.1.3.2 b)).

9.2 Formes de codage des chaînes

Les chaînes binaires, les chaînes d'octets et les chaînes de caractères restreintes sont codées sous forme primitive lorsqu'elles ne comportent pas plus de 1000 octets de contenu, et sinon, sous forme structurée. Les fragments de chaîne compris dans cette structure seront codés sous forme primitive. Le codage de chaque fragment, à l'exception éventuellement du dernier, comportera 1000 octets de contenu (et non comme au 8.20.6).

9.3 Eléments d'ensemble

Les codages des valeurs d'éléments d'un ensemble apparaîtront dans un ordre déterminé par leurs étiquettes conformément aux dispositions du 6.4 de la Rec. UIT-T X.680 | ISO/CEI 8824-1. De plus, pour déterminer l'ordre de codage des composantes d'un type "choix" non étiqueté, chacun de ces types se verra attribuer un rang comme s'il possédait une étiquette égale à la plus petite étiquette apparaissant dans ce type de choix ou dans n'importe lequel des types de choix non étiquetés qui y sont imbriqués.

Exemple – Dans ce qui suit, on suppose un environnement d'étiquetage implicite IMPLICIT TAGS:

```
A ::= SET
{
  a    [3] INTEGER,
  b    [1] CHOICE
  {
    c    [2] INTEGER,
    d    [4] INTEGER
  },
  e    CHOICE
  {
    f    CHOICE
    {
      g    [5] INTEGER,
      h    [6] INTEGER
    },
    i    CHOICE
    {
      j    [0] INTEGER
    }
  }
}
```

L'ordre dans lequel les éléments de l'ensemble sont codés est toujours e, b, a, puisque l'étiquette [0] est la plus petite, suivie de [1] puis [3].

10 Règles de codage distinctives

Le codage distinctif d'une valeur de données correspond au codage de cette valeur selon les règles du codage de base décrites à l'article 8, avec les restrictions suivantes ainsi que les restrictions énumérées à l'article 11.

NOTE – La Rec. X.509 | ISO/CEI 9594-8 interdit l'utilisation de valeurs abstraites en base 10 dans les applications d'annuaires.

10.1 Formes de longueur

Le codage utilisera la forme de longueur définie, codée sur le nombre minimal d'octets (en opposition avec 8.1.3.2 b)).

10.2 Formes de codage des chaînes

On n'utilisera pas la forme structurée pour coder les chaînes binaires, les chaînes d'octets et les chaînes de caractères restreintes (en opposition avec 8.20.6).

10.3 Éléments d'ensemble

Les codages des valeurs d'éléments d'un ensemble apparaîtront dans un ordre déterminé par leurs étiquettes selon les spécifications du 6.4 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

NOTE – Lorsqu'un élément d'un ensemble est d'un type choix non étiqueté, son rang de codage dépendra de l'étiquette de l'élément du choix effectivement codé.

11 Restrictions aux règles de codage de base applicables aux règles de codage canoniques et distinctives

L'indication "est le codage de base" paraissant dans les différents points de l'article 8 sera interprétée comme "est le codage canonique ou distinctif selon ce qui convient" (voir 8.16.1, 8.17.6, 8.17.8, 8.18.1, 8.21.6 et 8.21.8).

11.1 Valeurs booléennes

Si le codage représente la valeur booléenne "Vrai", son unique octet de contenu aura ses huit bits à 1 (en opposition avec 8.2.2).

11.2 Bits inutilisés

11.2.1 Chaque bit inutilisé dans l'octet final du codage d'une chaîne binaire sera mis à 0.

11.2.2 Là où les dispositions du 19.7 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 s'appliquent, les bits de fin à 0 de la chaîne binaire seront supprimés avant codage.

NOTES

1 Dans le cas où la contrainte de taille a été appliquée, la valeur abstraite fournie par le codeur à l'application sera une de celles satisfaisant la contrainte de taille et ne différant de la valeur transmise que par le nombre de bits de fin à 0.

2 Si une valeur de chaîne binaire ne comporte pas de bits à 1, le codeur générera une valeur de longueur nulle sans champ de contenu.

11.3 Valeurs réelles

11.3.1 Si le codage représente une valeur réelle en base $B = 2$, on utilisera un codage binaire en base 2. Avant de procéder au codage, la mantisse M et l'exposant E seront choisis de telle sorte que M soit nulle ou impaire.

NOTE – Ceci est une nécessité, car la même valeur réelle peut être tout aussi bien écrite $\{M, 2, E\}$ que $\{M', 2, E'\}$ avec $M \neq M'$ s'il existe un entier n différent de zéro tel que:

$$M' = M \times 2^{-n}$$

$$E' = E + n$$

Remplacée par une version plus récente ISO/CEI 8825-1 : 1995 (F)

Lors du codage de la valeur, le facteur d'échelle binaire F sera nul et M et E seront codés chacun sur le plus petit nombre d'octets nécessaires.

11.3.2 Si le codage représente une valeur réelle en base $B = 10$, on utilisera un codage décimal. Lors du codage, les dispositions suivantes s'appliqueront.

11.3.2.1 La forme NR3 de ISO 6093 sera utilisée (voir 8.5.6).

11.3.2.2 Le caractère d'espace SPACE ne sera pas utilisé en codage.

11.3.2.3 Si la valeur réelle est négative, son codage commencera par un signe moins (-), sinon il commencera par un chiffre.

11.3.2.4 Ni le premier ni le dernier chiffre de la mantisse ne doivent être égaux à 0.

11.3.2.5 Le dernier chiffre de la mantisse sera immédiatement suivi d'un point (.), suivi par la marque d'exponentiation "E".

11.3.2.6 Si l'exposant a la valeur 0, il sera écrit "+0", sinon le premier chiffre de l'exposant sera non nul, et le signe (+) ne sera pas utilisé.

11.4 Valeurs du type chaîne générale GeneralString

Le codage des valeurs du type chaîne générale "GeneralString" (et de ses sous-types) ne génère des séquences d'échappement pour désigner et invoquer une nouvelle entrée de registre que lorsque l'entrée de registre pour le caractère diffère du registre courant désigné par G0, C0 ou C1. Toutes les désignations et invocations se trouveront dans l'ensemble G0 ou C0.

NOTE – On suppose que chaque caractère d'une chaîne de caractères est associé à une entrée particulière dans le registre international des jeux de caractères codés.

11.5 Eléments d'ensemble et éléments de séquence avec valeur par défaut

La valeur d'une composante d'un ensemble ou d'une séquence ne sera pas codée si elle est égale à sa valeur par défaut.

11.6 Eléments d'ensemble-de

Les codages des éléments d'un ensemble-de apparaîtront en ordre ascendant, les codages étant comparés en tant que chaînes d'octets.

11.7 Temps généralisé

11.7.1 Le codage se terminera par un "Z", comme indiqué dans la disposition relative au temps généralisé de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

11.7.2 Le codage de la partie fractionnaire de seconde éliminera tous les 0 à droite, et si cette partie fractionnaire est nulle, ces éléments seront tous éliminés de même que le séparateur décimal.

Exemple – Un temps en secondes de "26.000" est représenté par "26", et un temps en secondes de "26.5200" par "26.52".

11.7.3 Le séparateur décimal, s'il est présent, sera le point ".".

11.7.4 Minuit (GMT) sera représenté sous la forme:

"AAAAMMJJ000000Z"

où "AAAAMMJJ" représente le jour suivant le minuit en question.

11.7.5 Exemples de représentations valides

"19920521000000Z"

"19920622123421Z"

"19920722132100.3Z".

11.7.6 Exemples de représentations non valides

"19920520240000Z" (représentation incorrecte de minuit)
 "19920622123421.0Z" (0 à droite parasite)
 "19920722132100.30Z" (0 à droite parasite).

12 Utilisation des règles de codage canoniques, distinctives et de base dans une définition de syntaxe de transfert

12.1 Il est possible de se référer aux règles de codage spécifiées dans la présente Recommandation | Norme internationale et de les appliquer chaque fois qu'il est nécessaire de spécifier une représentation en chaîne d'octets non ambiguë, non divisée et autodélimitante de toutes les valeurs d'un type unique ASN.1.

NOTE – Toutes les chaînes d'octets de ce type sont non ambiguës dans le cadre d'un type ASN.1 unique, mais elles ne le seraient pas nécessairement si elles étaient mêlées à des codages d'un type ASN.1 différent.

12.2 L'identificateur d'objet et le descripteur d'objet suivants sont affectés à l'identification et à la description des règles de codage de base spécifiées dans la présente Recommandation | Norme internationale:

```
{joint-iso-ccitt asn1 (1) basic-encoding (1)}
```

et

"Basic Encoding of a single ASN.1 type"
 (codage de base d'un type ASN.1 unique)

12.3 L'identificateur d'objet et le descripteur d'objet suivants sont affectés à l'identification et à la description des règles de codage canoniques spécifiées dans la présente Recommandation | Norme internationale:

```
{joint-iso-ccitt asn1(1) ber-derived(2) canonical-encoding(0)}
```

et

"Canonical encoding of a single ASN.1 type"
 (codage canonique d'un type ASN.1 unique)

12.4 L'identificateur d'objet et le descripteur d'objet suivants sont affectés à l'identification et à la description des règles de codage distinctives spécifiées dans la présente Recommandation | Norme internationale:

```
{joint-iso-ccitt asn1(1) ber-derived(2) distinguished-encoding(1)}
```

et

"Distinguished encoding of a single ASN.1 type"
 (codage distinctif d'un type ASN.1 unique)

12.5 Lorsqu'une spécification non ambiguë définit une syntaxe abstraite comme un ensemble de valeurs de données de présentation, dont chacune est une valeur d'un type ASN.1 donné spécifiquement nommé – en général (mais pas nécessairement) un type choix – une des valeurs d'identificateur d'objet spécifiées aux 12.2, 12.3 ou 12.4 peut alors être utilisée avec le nom de la syntaxe abstraite pour identifier respectivement les règles de codage de base, canoniques ou distinctives appliquées au codage du type ASN.1 spécifiquement nommé utilisé dans la définition de la syntaxe abstraite.

12.6 Les noms spécifiés aux 12.2, 12.3 et 12.4 ne seront utilisés avec un nom de syntaxe abstraite pour identifier une syntaxe de transfert, que si les conditions énoncées au 12.5 pour la définition de la syntaxe abstraite sont remplies (voir D.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1).

Annexe A

Exemples de codages

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

Cette annexe illustre les règles de codage de base spécifiées dans la présente Recommandation | Norme internationale, en montrant la représentation en octets d'un enregistrement "salarié" (fictif) qui est défini en ASN.1.

A.1 Description ASN.1 de la structure de l'enregistrement

La structure de l'enregistrement fictif (salarié) est décrite formellement ci-dessous, en utilisant la notation ASN.1 spécifiée dans la Rec. X.680 | ISO/CEI 8824-1 pour définir les types.

```

PersonnelRecord ::= [APPLICATION 0] IMPLICIT SET {
    name          Name,
    title         [0] VisibleString,
    number        EmployeeNumber,
    dateOfHire    [1] Date,
    nameOfSpouse  [2] Name,
    children      [3] IMPLICIT
        SEQUENCE OF ChildInformation DEFAULT {} }

ChildInformation ::= SET
    { name      Name,
      dateOfBirth [0] Date}

Name ::= [APPLICATION 1] IMPLICIT SEQUENCE
    {givenName  VisibleString,
     initial    VisibleString,
     familyName VisibleString}

Matricule-salarié ::= [APPLICATION 2] IMPLICIT INTEGER

Date ::= [APPLICATION 3] IMPLICIT Chaîne-visible -- AAAAMMJJ
    
```

A.2 Description ASN.1 d'une valeur d'enregistrement

La valeur de l'enregistrement "salarié" de John P. Smith est décrite formellement ci-dessous en ASN.1.

```

{ name {givenName "John",initial "P",familyName "Smith"},
  title "Director",
  number 51,
  dateOfHire "19710917",
  nameOfSpouse {givenName "Mary",initial "T",familyName "Smith"},
  children
    {{{givenName "Ralph",initial "T",familyName "Smith"},
      dateOfBirth "19571111"},
     {{givenName "Susan",initial "B",familyName "Jones"},
      dateOfBirth "19590717"}}}
    
```

A.3 Représentation de la valeur de cet enregistrement

La représentation en octets de la valeur d'enregistrement donnée ci-dessus (après application des règles de codage de base définies dans la présente Recommandation | Norme internationale) est présentée ci-après. Les valeurs des champs d'identification, de longueur et des contenus numériques sont indiquées en hexadécimal, à raison de deux chiffres par octet. Les valeurs du contenu des chaînes de caractères sont représentées sous forme textuelle, à raison d'un caractère par octet.

| Enreg. | Longueur | Contenu | |
|---------|----------------|----------|---------|
| Salarié | 60 | 8186 | |
| | Nom | Longueur | Contenu |
| | 61 | 10 | |
| | Chaîne visible | Longueur | Contenu |
| | 1A | 04 | "John" |
| | Chaîne visible | Longueur | Contenu |
| | 1A | 01 | "P" |

| | | | | | | | |
|-----------------|----------------|----------------------|----------------|------------------------|----------------|-----------------------|----------------|
| | | Chaîne visible 1A | Longueur 05 | Contenu "Smith" | | | |
| Fonction A0 | Longueur 0B | Contenu | | | | | |
| | | Chaîne visible 1A | Longueur 09 | Contenu "Directeur" | | | |
| Matricule 42 | Longueur 01 | Contenu 33 | | | | | |
| Embauche A1 | Longueur 0A | Contenu | | | | | |
| | | Date 43 | Longueur 08 | Contenu "19710917" | | | |
| Conjoint A2 | Longueur 12 | Contenu | | | | | |
| | | Nom 61 | Longueur 10 | Contenu | | | |
| | | | | Chaîne visible 1A | Longueur 04 | Contenu "Mary" | |
| | | | | Chaîne visible 1A | Longueur 01 | Contenu "T" | |
| | | | | Chaîne visible 1A | Longueur 05 | Contenu "Smith" | |
| [3] A3 | Longueur 42 | Contenu | | | | | |
| | | Ensemble 31 | Longueur 1F | Contenu | | | |
| | | | | Nom 61 | Longueur 11 | Contenu | |
| | | | | | | Chaîne visible 1A | Longueur 05 |
| | | | | | | Contenu "Ralph" | |
| | | | | | | Chaîne visible 1A | Longueur 01 |
| | | | | | | Contenu "T" | |
| | | | | | | Chaîne visible 1A | Longueur 05 |
| | | | | | | Contenu "Smith" | |
| | | | | Naissance A0 | Longueur 0A | Contenu | |
| | | | | | | Date 43 | Longueur 08 |
| | | | | | | Contenu "19571111" | |
| | | Ensemble 31 | Longueur 1F | Contenu | | | |
| | | | | Nom 61 | Longueur 11 | Contenu | |
| | | | | | | Chaîne visible 1A | Longueur 05 |
| | | | | | | Contenu "Susan" | |
| | | | | | | Chaîne visible 1A | Longueur 01 |
| | | | | | | Contenu "B" | |
| | | | | | | Chaîne visible 1 | Longueur 05 |
| | | | | | | Contenu "Jones" | |
| | | | | Naissance A0 | Longueur 0A | Contenu | |
| | | | | | | Date 43 | Longueur 08 |
| | | | | | | Contenu "19590717" | |

Annexe B

Affectation des valeurs d'identificateur d'objet

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

Les valeurs suivantes d'identificateur et de descripteur d'objet sont affectées par la présente Recommandation | Norme internationale.

Référence **Valeur d'identificateur d'objet**

12.2 {joint-iso-ccitt asn1 (1) basic-encoding (1)}

Valeur de descripteur d'objet

"Basic Encoding of a single ASN.1 type"
(codage de base d'un type ASN.1 unique)

Référence **Valeur d'identificateur d'objet**

12.3 {joint-iso-ccitt asn1(1) ber-derived(2) canonical-encoding(0)}

Valeur de descripteur d'objet

"Canonical encoding of a single ASN.1 type"
(codage canonique d'un type ASN.1 unique)

Référence **Valeur d'identificateur d'objet**

12.4 {joint-iso-ccitt asn1(1) ber-derived(2) distinguished-encoding(1)}

Valeur de descripteur d'objet

"Distinguished encoding of a single ASN.1 type"
(codage distinctif d'un type ASN.1 unique)

Annexe C

Illustration du codage d'une valeur réelle

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

C.1 Un expéditeur déterminera normalement, d'après la représentation des nombres en virgule flottante sur son propre matériel, les algorithmes (indépendants de la valeur) à utiliser pour transférer des valeurs entre cette représentation en virgule flottante et les champs de longueur et de contenu du codage ASN.1 d'un réel. A partir de la représentation (fictive) en virgule flottante de la mantisse représentée à la Figure C.1, cette annexe illustre la démarche à suivre.

Il est supposé que l'exposant peut facilement être obtenu à partir de la représentation en virgule flottante de la machine, sous la forme d'un entier E.

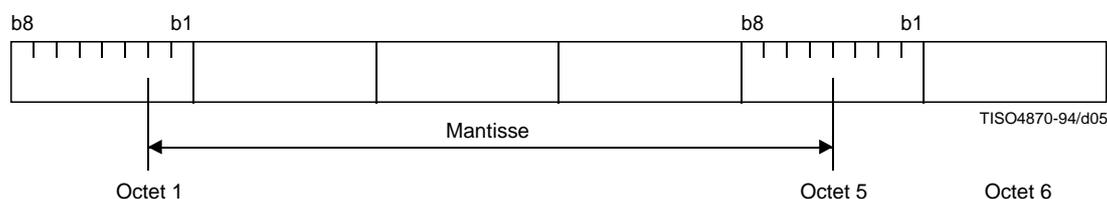


Figure C.1 – Représentation d'un réel en virgule flottante

C.2 Le champ de contenu qui doit être généré pour expédier une valeur non nulle en codage binaire (tel qu'il est spécifié dans le corps de la présente Recommandation | Norme internationale) est constitué des éléments suivants:

1 S bb ff ee Champ de E Champ de N

où S (signe de la mantisse) dépend de la valeur à convertir, bb est une valeur fixe (disons 10) représentant la base (dans ce cas on suppose une base 16), ff est la valeur fixe F calculée comme le décrit C.3, et ee est une valeur fixe de longueur d'exposant calculée comme le décrit C.4 (cette annexe ne traite pas le cas où E nécessite plus de 3 octets).

C.3 L'algorithme transmettra les octets 1 à 5 de la représentation interne à la machine comme valeur de N, après avoir forcé à 0 les bits 8 à 3 de l'octet 1 et les bits 4 à 1 de l'octet 5. Le point décimal implicite est supposé être positionné entre les bits 2 et 1 de l'octet 1 dans la représentation machine qui fournit la valeur de E. Sa position implicite peut être décalée au premier point après la fin de l'octet 5 en réduisant la valeur de E avant transmission. Dans le système de notre exemple, nous pouvons décaler de quatre bits pour chaque décrémentation de l'exposant (parce que nous avons supposé une base 16) en sorte qu'une décrémentation de 9 positionnera le point implicite entre les bits 6 et 5 de l'octet 6. Pour positionner correctement le point à M, la valeur de M est N multipliée par 2^3 (la position implicite dans N, après transfert des octets, se trouve après le bit 1 de l'octet 5). Nous avons ainsi les paramètres cruciaux suivants:

$$F = 3 \quad (\text{et donc } ff = 11)$$

$$\text{décrémentation de l'exposant} = 9$$

C.4 La longueur nécessaire à l'exposant est alors calculée en prenant le nombre maximal d'octets nécessaires pour représenter les valeurs:

$$E_{\min} - \text{excès} - \text{décrémentation de l'exposant}$$

$$E_{\max} - \text{excès} - \text{décrémentation de l'exposant}$$

où E_{\min} et E_{\max} sont les valeurs entières minimale et maximale de la représentation de l'exposant, et où l'excès est la valeur quelconque qui doit être soustraite pour donner la vraie valeur de l'exposant, et la décrémentation de l'exposant est celle qui est calculée au C.3. Supposons que ceci donne une longueur de trois octets, alors $ee = 10$. Supposons également que l'excès soit nul.

C.5 L'algorithme de transmission devient alors:

- a) transmettre le champ d'identificateur (défini par les règles de codage de base) avec une étiquette de réel de type ASN.1;
- b) examiner si la valeur est nulle; si oui transmettre un champ de longueur (défini par les règles de codage de base) de valeur nulle (aucun octet de contenu) et terminer l'algorithme;
- c) déterminer et enregistrer le signe de la mantisse, et prendre l'opposé de la mantisse si elle est négative;
- d) transmettre un champ de longueur (défini par les règles de codage de base ASN.1) de valeur 9; on aura alors:
11101110, si la valeur est négative,
ou 10101110, si la valeur est positive;
- e) générer et transmettre l'exposant à trois octets, avec la valeur
 $E - 9$
- f) mettre à zéro les bits 8 à 3 de l'octet 1 et les bits 4 à 1 de l'octet 5, puis transmettre la mantisse à 5 octets.

C.6 L'algorithme du destinataire doit pouvoir traiter n'importe quel format ASN.1, mais ici, la valeur en virgule flottante peut être directement utilisée. Procéder comme suit:

- a) examiner l'octet 1 du contenu; s'il est égal à 1x101110, la transmission est compatible avec la représentation du destinataire: il suffit d'inverser l'algorithme de l'émetteur;
- b) autrement, pour le codage des caractères, appeler le logiciel de conversion du codage décimal par caractère standard en virgule flottante, et traiter la valeur réelle spéciale "SpecialRealValue" conformément à la sémantique de l'application (peut-être en définissant les plus grand et plus petit nombres en virgule flottante pouvant être représentés par la machine);
- c) pour une transmission binaire, entrer N dans l'unité de virgule flottante, en éliminant si nécessaire les octets de plus faible poids; multiplier par 2^F et par B^E , puis prendre l'opposé s'il y a lieu. Les réalisateurs trouveront dans certains cas des optimisations possibles, mais ils constateront que de telles optimisations seront plus coûteuses en tests qu'elles ne leur rapporteront (sauf dans le cas évident de transmission entre machines compatibles).

C.7 Les algorithmes ci-dessus sont fournis à titre indicatif. Il appartient bien entendu aux réalisateurs de déterminer les stratégies les plus avantageuses.

Annexe D

Utilisation des règles de codage distinctives (DER) et canoniques (CER) en authentification d'origine des données

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

D.1 Problème à résoudre

D.1.1 Les règles de codage distinctives et les règles de codage canoniques ont été établies pour aider à créer des mécanismes de sécurité d'intégrité utilisant des authentificateurs pour les valeurs à transférer.

NOTE – Pour plus de clarté, il ne sera fait état que des seules règles de codage distinctives dans le reste de cette annexe. Mais le contenu s'applique aussi bien aux règles de codage canoniques.

D.1.2 Le concept d'authentificateur est assez trivial: il consiste à prendre la suite binaire à transférer, à lui appliquer une certaine fonction de hachage pour la réduire à quelques octets, à chiffrer ces octets sous la forme d'un authentificateur pour en authentifier l'origine, puis à transmettre cet authentificateur avec la valeur initiale à transférer (laquelle est transmise en clair). Dès qu'il est reçu, l'authentificateur est recalculé à partir du texte reçu en clair et comparé avec l'authentificateur reçu; s'ils sont égaux, le texte n'a pas été trafiqué, sinon, il l'a été.

D.1.3 Ce concept simple se complique dans une architecture ISO et, notamment, lorsque la couche présentation est utilisée.

D.1.4 Deux problèmes se posent, dont l'un est un problème de modélisation et a trait à l'indépendance des couches, et l'autre concerne l'utilisation de relais dans la couche application, comme ceux qui sont utilisés dans la Recommandation X.400.

D.1.5 En ce qui concerne le problème de modélisation, la fonction de hachage et l'algorithme de chiffrement font partie du fonctionnement de l'application, mais cette dernière n'a pas la connaissance ou la maîtrise du codage effectif qu'utilisera la couche présentation. De même, à la réception, le décodage et, donc, la restructuration de la chaîne binaire relèvent de la couche Présentation. Quatre solutions ont été proposées pour résoudre ce problème:

- a) condamner l'utilisation dans l'authentificateur des octets produits par la couche Présentation (point de vue actuel adopté par les spécialistes de la couche Présentation et le groupe ULA);
- b) mettre les mécanismes de hachage et d'authentification dans la couche Présentation même (solution rejetée dans le cadre de la question générale de la prise en charge du chiffrement dans l'ASN.1; au moment où cette décision a été prise, la raison invoquée était que les travaux sur la sécurité n'étaient pas suffisamment avancés et qu'on ne voulait pas préjuger du résultat);
- c) modéliser une interaction complexe avec la couche Présentation dans laquelle les valeurs à transmettre sont présentées à la couche Présentation pour être codées, qui les code et les renvoie à la couche Application, qui calcule l'authentificateur et renvoie le tout en transmission; en réception, et parallèlement à la régénération de la valeur abstraite, la signature codée reçue est passée à la couche Application pour en vérifier l'authentificateur (ce modèle a été rejeté par le groupe ULA);
- d) réaliser tout le codage dans la couche Application et ne pas utiliser les services de présentation pour la négociation de la syntaxe de transfert (cette solution contredit tout à fait l'esprit du modèle de référence OSI et ne peut être acceptée comme solution générale).

D.1.6 On pourrait penser que le fait de ne pas parvenir à un accord sur un modèle pour décrire un processus apparemment simple et faisable (générer le message codé, puis l'authentificateur, transmettre les deux et vérifier l'authentificateur à la réception) n'est pas acceptable à long terme. Ce point de vue serait tout à fait fondé n'était-ce l'existence du second problème des relais de la couche Application et s'il n'y avait pas d'autre solution faisable. (La présente annexe décrit une solution de remplacement, déjà utilisée dans la Rec. X.509 du CCITT | ISO/CEI 9594-8, qui ne pose pas de problèmes de modélisation ou de relais et qui est considérée comme faisable.)

D.1.7 Le second problème est que, si un relais d'application est en place, la syntaxe de transfert utilisée pour la deuxième transmission peut être différente de la syntaxe adoptée pour la première (par exemple, utilisation des règles DER sur l'une et BER sur l'autre). Ce changement mettrait en défaut l'authentificateur, à moins que celui-ci ne soit décodé et recalculé au relais, ce qui supposerait la sûreté des échanges avec ce dernier alors que ce qui est demandé c'est une sécurité de bout en bout.

NOTE – Il a été suggéré de marquer le contexte de présentation par un indicateur pas de décodage/recodage aux relais d'Application, mais cette solution pose elle aussi entre autres divers problèmes de modélisation.

D.1.8 Nous sommes donc amenés à essayer de travailler avec un modèle dans lequel la couche Présentation (et les éventuels relais d'application) assure le transfert de la syntaxe abstraite et de la sémantique de l'information, mais ne garantit pas que le codage binaire effectif (syntaxe de transfert) soit le même de bout en bout.

D.1.9 La gageure est donc de créer un mécanisme d'authentification qui puisse intervenir sur le type de données abstraites et non sur la chaîne binaire transmise.

D.1.10 Le groupe de travail sur l'Annuaire a été le premier à essayer de trouver une solution à ce problème, et c'est son modèle qui est présenté dans ce qui suit.

D.2 Approche de la solution

D.2.1 Le texte suivant décrit d'abord un modèle conceptuel de ce qui est fait, puis une réalisation optimisée qui élimine le double codage/décodage que suppose le modèle.

D.2.2 Le modèle conceptuel fonctionne de la manière suivante:

- a) L'expéditeur, situé dans la couche application, convertit la valeur de syntaxe abstraite en une chaîne binaire par les règles de codage distinctives (DER), et génère l'authentificateur à partir de cette chaîne; ces deux valeurs sont alors transmises par les mécanismes normaux de la couche présentation avec une éventuelle syntaxe de transfert. Conceptuellement, l'expéditeur procède à un double codage, une fois pour l'authentificateur (en utilisant les règles DER) dans la couche Application et une autre fois pour le transfert proprement dit (au moyen de la syntaxe de transfert négociée) dans la couche Présentation.

NOTE – Une propriété importante de la chaîne binaire générée par les règles DER est d'être en correspondance biunivoque avec la valeur abstraite. Le transfert de bout en bout sans perte d'information au niveau de la syntaxe abstraite équivaut donc à un transfert de bout en bout de la chaîne binaire à partir de laquelle est calculé l'authentificateur.

- b) Le destinataire décode la chaîne binaire reçue dans la couche Présentation, en utilisant la syntaxe de transfert négociée (qui peut différer de celle utilisée par l'expéditeur si un relais d'application est mis en jeu) et passe la valeur abstraite à l'application. Dans la couche application, la valeur abstraite est recodée à l'aide des règles DER pour générer la chaîne binaire à authentifier.

D.2.3 Par conséquent, du point de vue conceptuel, nous procédons à deux codages côté émission, puis à un décodage et à un codage côté réception. Les développeurs peuvent choisir de procéder de cette manière si le fournisseur du code mis en œuvre dans la couche présentation est différent du fournisseur du code de l'application. On ne connaît pas au stade actuel la charge supplémentaire que représenterait une telle procédure. Toutefois, dans le cadre d'une réalisation intégrée, il est possible d'optimiser le système de la manière décrite ci-dessous. Il convient en outre de noter que les règles distinctives ne sont pas plus difficiles à appliquer que les règles de base, sauf en ce qui concerne l'utilisation des "ensemble-de". En effet, si un ensemble-de important doit être traité, il se peut qu'il faille recourir à un programme de tri sur disque. Les concepteurs d'applications doivent garder ce problème à l'esprit, et essayer d'utiliser des "séquence-de" plutôt que des "ensemble-de" lorsqu'ils envisagent d'employer les règles de codage distinctives.

D.3 Optimisation du produit

D.3.1 Si le modèle OSI et les normes de protocole spécifient un comportement donné, ils ne cherchent en aucun cas à contraindre l'architecture et la structure du code de la réalisation même. Un développeur peut donc obtenir ce comportement de la manière qu'il choisira.

D.3.2 Côté émission, la chaîne binaire générée (conceptuellement dans la couche Application) peut être conservée et utilisée pour effectuer le codage conceptuellement exécuté dans la couche Présentation. Ceci convient pour l'émission si la syntaxe de transfert négociée correspond aux règles de codage distinctives (DER) ou de base (BER) de l'ASN.1; si ce n'est pas le cas, le double codage s'impose.

D.3.3 De même, du côté réception, la chaîne binaire reçue peut être conservée (pour toute syntaxe de transfert) et la réalisation peut l'utiliser pour vérifier l'authentificateur. S'il y a correspondance, le problème est réglé, sinon, il peut alors s'agir d'un problème de syntaxe de transfert, auquel cas il faut recommencer le codage à partir de la valeur abstraite pour déterminer si les valeurs ont été ou non trafiquées.

D.3.4 Pour maximiser les chances de ne pas avoir à procéder à un double codage/décodage, il serait souhaitable que les systèmes utilisant ce mécanisme essaient de négocier une syntaxe de transfert relevant des règles de codage distinctives (au moyen de l'identificateur d'objet approprié) comme choix préférentiel, avec repli sur les règles de base (premier repli) puis sur des règles de codage quelconques (deuxième repli).