



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**X.208**

**INTERCONNEXION DES SYSTÈMES OUVERTS  
MODÈLE ET NOTATION**

---

**SPÉCIFICATION DE LA SYNTAXE ABSTRAITE  
NUMÉRO UN (ASN.1)**

**Recommandation UIT-T X.208**

(Extrait du *Livre Bleu*)

---

## NOTES

1 La Recommandation X.208 de l'UIT-T a été publiée dans le fascicule VIII.4 du Livre Bleu. Ce fichier est un extrait du Livre Bleu. La présentation peut en être légèrement différente, mais le contenu est identique à celui du Livre Bleu et les conditions en matière de droits d'auteur restent inchangées (voir plus loin).

2 Dans la présente Recommandation, le terme «Administration» désigne indifféremment une administration de télécommunication ou une exploitation reconnue.

© UIT 1988, 1993

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.

## Recommandation X.208

### SPÉCIFICATION DE LA SYNTAXE ABSTRAITE NUMÉRO UN (ASN.1)<sup>1)</sup>

(Melbourne, 1988)

Le CCITT,

*considérant*

- (a) la diversité et la complexité des objets informationnels véhiculés dans la Couche Application;
- (b) la nécessité d'une notation de haut niveau pour spécifier de tels objets informationnels;
- (c) l'intérêt d'une formulation et d'une normalisation des règles de codage de ces objets informationnels,

*recommande à l'unanimité*

- (1) la notation à utiliser pour définir la syntaxe abstraite des objets informationnels spécifiée dans la section 1;
- (2) les types de chaînes de caractères définis dans la section 2;
- (3) les autres types utiles définis dans la section 3;
- (4) les sous-types définis dans la section 4.

### SOMMAIRE

0	<i>Introduction</i>
1	<i>Objet et domaine d'application</i>
2	<i>Références</i>
3	<i>Définitions</i>
4	<i>Abréviations</i>
5	<i>Notation utilisée dans la présente Recommandation</i>
5.1	Productions
5.2	Partie droite d'une production
5.3	Exemple de production
5.4	Disposition
5.5	Récurtivité
5.6	Dénotation d'un ensemble de séquences
5.7	Références à un item
5.8	Etiquettes
6	<i>Utilisation de la notation ASN.1</i>

---

<sup>1)</sup> La Recommandation X.208 et la norme ISO 8824, "Systèmes de traitement de l'information – Interconnexion des systèmes ouverts - Spécification de la syntaxe abstraite numéro un (ASN.1)", telle que développée par son addendum 1, ont été établies en étroite collaboration et sont techniquement alignées.

## SECTION 1 – SPÉCIFICATION DE LA NOTATION ASN.1

- 7 *Le jeu de caractères ASN.1*
- 8 *Items ASN.1*
  - 8.1 Règles générales
  - 8.2 Dénotations de types
  - 8.3 Identificateurs
  - 8.4 Dénotations de valeur
  - 8.5 Dénotations d'un module
  - 8.6 Commentaire
  - 8.7 Item vide
  - 8.8 Item nombre
  - 8.9 Item chaîne binaire
  - 8.10 Item chaîne hexadécimale
  - 8.11 Items chaîne de caractères
  - 8.12 Item affectation
  - 8.13 Items caractère unique
  - 8.14 Items mot clé
- 9 *Définition d'un module*
- 10 *Dénotation des définitions de types et de valeurs*
- 11 *Affectation de types et de valeurs*
- 12 *Définition des types et valeurs*
- 13 *Notation du type Booléen*
- 14 *Notation du type Entier*
- 15 *Notation du type Enuméré*
- 16 *Notation du type Réel*
- 17 *Notation du type Chaîne binaire*
- 18 *Notation du type Chaîne d'octets*
- 19 *Notation du type Vide*
- 20 *Notation des types Séquence*
- 21 *Notation des types Séquence-de*
- 22 *Notation des types Ensemble*

- 23 *Notation des types Ensemble-de*
- 24 *Notation des types Choix*
- 25 *Notation des types Sélection*
- 26 *Notation des types Etiquetés*
- 27 *Notation du type Quelconque*
- 28 *Notation du type Identificateur d'objet*
- 29 *Notation des types Chaîne de caractères*
- 30 *Notation des types définis dans la section 3*

## SECTION 2 – TYPES CHAÎNE DE CARACTÈRES

- 31 *Définition des types Chaîne de caractères*

## SECTION 3 – DÉFINITIONS UTILES

- 32 *Heure généralisée*
- 33 *Heure universelle*
- 34 *Le type Externe*
- 35 *Le type Descripteur d'objet*

## SECTION 4 – SOUS-TYPES

- 36 *Notation des sous-types*
- 37 *Ensembles de valeurs d'un sous-type*
  - 37.1 *Valeur unique*
  - 37.2 *Sous-type contenu*
  - 37.3 *Plage de valeurs*
  - 37.4 *Limitation de taille*
  - 37.5 *Alphabet autorisé*
  - 37.6 *Sous-typage interne*

## *Annexe A – La notation de macro*

- A.1 *Introduction*
- A.2 *Extensions du jeu de caractères et des items ASN.1*
  - A.2.1 *Dénotation-macro*
  - A.2.2 *Dénotation-production*
  - A.2.3 *Dénotation-type-Local*

- A.2.4 Dénotation-valeur-locale
- A.2.5 Item de sélection
- A.2.6 Item de terminaison de définition
- A.2.7 Item terminal syntaxique
- A.2.8 Items mots clés de catégorie syntaxique
- A.2.9 Items mots clés additionnels
- A.3 Notation des définitions de macro
- A.4 Utilisation de la nouvelle notation

*Annexe B* – Affectation par l'ISO des valeurs des composants d'un *identificateur d'objet*

*Annexe C* – Affectation par le CCITT des valeurs des composants d'un *identificateur d'objet*

*Annexe D* – Affectation commune des valeurs des composants d'un *identificateur d'objet*

*Appendice I* – Exemples et directives

- I.1 Exemple d'un enregistrement "salarié"
  - I.1.1 Description informelle de l'enregistrement "salarié"
  - I.1.2 Description ASN.1 de la structure d'enregistrement
  - I.1.3 Description ASN.1 d'une valeur de l'enregistrement
- I.2 Directives relatives à l'utilisation de la notation
  - I.2.1 Booléen
  - I.2.2 Entier
  - I.2.3 Enuméré
  - I.2.4 Réel
  - I.2.5 Chaîne binaire
  - I.2.6 Chaîne d'octets
  - I.2.7 Vide
  - I.2.8 Séquence et Séquence-de
  - I.2.9 Ensemble
  - I.2.10 Etiqueté
  - I.2.11 Choix
  - I.2.12 Type Sélection
  - I.2.13 Quelconque
  - I.2.14 Externe
  - I.2.15 Codé
- I.3 Exemple d'utilisation de la notation de macro
- I.4 Utilisation d'ASN.1 pour l'identification des syntaxes abstraites
- I.5 Sous-types

*Appendice II* – Résumé de la notation ASN.1

## 0 Introduction

Dans les couches inférieures du modèle de référence de base (voir la Recommandation X.200) chaque paramètre "données de l'utilisateur" d'une primitive de service est spécifié comme la valeur binaire d'une séquence d'octets.

Dans la Couche Présentation, la nature des paramètres "données de l'utilisateur" est différente. Les spécifications de la Couche Application imposent aux données de l'utilisateur du service de présentation (voir la Recommandation X.216) de véhiculer les valeurs de types très complexes, incluant éventuellement des chaînes de caractères de divers jeux. Pour spécifier la valeur véhiculée, ces recommandations exigent une notation définie qui ne détermine pas la représentation de la valeur. Ces recommandations sont complétées par la spécification d'un ou plusieurs algorithmes appelés **règles de codage** qui déterminent les valeurs des octets de la Couche Session véhiculant ces valeurs de la Couche Application (appelées **syntaxes de transfert**). Le protocole de la Couche Présentation (voir Recommandation X.226) peut négocier les syntaxes de transfert à utiliser.

L'objet de la spécification d'une valeur est de permettre de la distinguer des autres valeurs possibles. L'ensemble constitué par la valeur et celles dont elle est distinguée, est appelé **un type**, dont une occurrence spécifique est une valeur de ce type. Plus généralement, une valeur ou un type peuvent souvent être considérés comme composés de plusieurs valeurs ou types plus simples, et de relations qui existent entre eux. L'expression **type de type de données** est souvent utilisée comme synonyme de type.

Pour interpréter correctement la représentation d'une valeur (qu'il s'agisse de signes sur du papier ou de bits sur une ligne de communication), il faut connaître (en général d'après le contexte) le type de la valeur représentée. L'identification d'un type est donc une partie importante de la présente Recommandation.

Une technique très générale pour définir un type compliqué est de définir un petit nombre de types simples, dont toutes les valeurs possibles sont définies, puis de combiner ces types simples de diverses façons. Voici certaines des façons de définir de nouveaux types:

- a) étant donné une liste (ordonnée) de types existants, une valeur peut être formée comme une séquence (ordonnée) de valeurs dont chacune est d'un des types existants; l'ensemble de toutes les valeurs possibles ainsi obtenu est d'un nouveau type (si les types de la liste sont tous distincts, ce mécanisme peut être étendu pour permettre l'omission de certaines valeurs de la liste);
- b) étant donné une liste de types existants (distincts) une valeur peut être formée comme un ensemble (non ordonné) de valeurs dont une de chacun des types existants, l'ensemble de toutes les valeurs possibles ainsi obtenu est un nouveau type; (là encore, le mécanisme peut être étendu pour permettre l'omission de certaines valeurs);
- c) étant donné un type existant, une valeur peut être formée comme une séquence (ordonnée) ou un ensemble (non ordonné) de zéro, une ou plusieurs valeurs du type existant; l'ensemble (infini) de toutes les valeurs possibles ainsi obtenu est un nouveau type;
- d) étant donné une liste de types (distincts) une valeur peut être choisie à partir de n'importe lequel d'entre eux; l'ensemble de toutes les valeurs possibles ainsi obtenu est un nouveau type;
- e) étant donné un type, un nouveau type peut être formé comme un sous-ensemble de ce type, induit par une certaine structure ou relation d'ordre entre les valeurs.

Les types ainsi définis sont appelés **types structurés**.

Une **étiquette** est affectée à chaque type défini à l'aide de la notation spécifiée dans la présente Recommandation. Cette étiquette est définie par cette Recommandation ou par l'utilisateur de la notation.

Il est courant d'attribuer la même étiquette à de nombreux types différents, chacun étant identifié par le contexte d'utilisation de cette étiquette.

L'utilisateur de la notation peut décider d'attribuer des étiquettes différentes à deux apparitions d'un même type, on crée ainsi deux types distincts. Ceci peut être nécessaire, quand il faut distinguer le choix qui a été opéré, dans des situations telles que celle de d) ci-dessus.

Quatre classes d'étiquettes sont spécifiées dans la notation.

La première est la classe **universelle**. Les étiquettes de la classe universelle sont uniquement utilisées comme spécifié dans la présente Recommandation, chaque étiquette étant affectée:

- a) soit à un type unique;
- b) soit à un mécanisme de construction.

La deuxième classe est la classe **application**. L'affectation à des types des étiquettes de la classe particulière à une application est spécifiée par d'autres normes ou recommandations. Dans une norme ou recommandation déterminée, une étiquette de la classe particulière à une application ne doit être affectée qu'à un seul type.

La troisième classe est la classe **privée**. Les étiquettes de la classe à usage privé ne sont jamais affectées par des normes internationales de l'ISO, ni par des Recommandations du CCITT. Leur utilisation est spécifique à une entreprise.

La dernière classe d'étiquettes est celle des étiquettes spécifiques à un contexte. Dans le cadre de toute application de cette notation, leur utilisation est libre et interprétée selon le contexte d'utilisation.

Les étiquettes sont principalement destinées à être utilisées par la machine et ne sont pas essentielles pour la notation lisible par l'homme, définie dans la présente Recommandation. Toutefois, lorsqu'il est impératif de distinguer certains types, cette distinction est exprimée en leur imposant des étiquettes distinctes. L'affectation des étiquettes est donc une partie importante de l'utilisation de la présente notation.

*Remarque 1* – Tous les types qui peuvent être définis dans la notation de la présente Recommandation ont une étiquette. Etant donné un certain type, l'utilisateur de la notation peut en définir un nouveau en lui donnant une étiquette différente.

*Remarque 2* – Les règles de codage associent toujours, explicitement ou implicitement, à toute représentation d'une valeur d'un type son étiquette. Les limitations imposées à l'utilisation de la notation, garantissent que l'étiquette est suffisante pour déterminer de façon non ambiguë le type véritable, sous réserve que les définitions de types applicables soient disponibles.

La présente Recommandation décrit une notation qui permet de définir des types complexes et de spécifier les valeurs de ces types, sans préjuger de la façon dont une occurrence de ce type doit être représentée lors du transfert (c'est-à-dire par quelle séquence d'octets). Une notation qui offre cette possibilité est appelée **notation de définition de syntaxe abstraite**.

L'objet de la présente Recommandation est de spécifier une notation de définition de syntaxe abstraite appelée **Notation de syntaxe abstraite numéro un** ou ASN.1. La Notation de syntaxe abstraite numéro un est employée comme outil semi-formel pour définir des protocoles. L'utilisation de la notation ne supprime pas automatiquement le risque d'ambiguïté des spécifications. Il appartient aux utilisateurs de la notation de s'assurer que leurs spécifications ne sont pas ambiguës.

La présente Recommandation est complétée par d'autres normes et recommandations qui spécifient des règles de codage. L'application des règles de codage à la valeur d'un type défini par ASN.1 se traduit par une spécification complète de la représentation des valeurs de ce type lors du transfert (syntaxe de transfert).

La présente Recommandation est alignée, sur le plan technique et rédactionnel, sur la Norme internationale ISO 8824 complétée par l'addendum 1 à cette norme.

La section 1 de la présente Recommandation définit les types simples pris en charge par ASN.1 et spécifie la notation à utiliser pour la dénotation de types simples et la définition de types structurés. La section 1 spécifie également la notation à utiliser pour spécifier les valeurs de types définies en ASN.1.

La section 2 de la présente Recommandation définit des types additionnels (types chaîne de caractères) qui peuvent, par application des règles de codage des jeux de caractères, être assimilés au type chaîne d'octets.

La section 3 de la présente Recommandation définit certains types structurés, considérés d'utilité générale et n'exigeant aucune règle de codage supplémentaire.

La section 4 de la présente Recommandation définit une notation qui permet d'obtenir des sous-types à partir des valeurs d'un type parent.

L'annexe A, qui fait partie de la présente Recommandation, spécifie une notation d'extension de la notation de base ASN.1 par la définition de macros.

L'annexe B, qui fait partie de la présente Recommandation, définit l'arbre des identificateurs d'objets du domaine de compétence de l'ISO.

L'annexe C, qui fait partie de la présente Recommandation, définit l'arbre des identificateurs d'objets du domaine de compétence du CCITT.

L'annexe D, qui fait partie de la présente Recommandation, définit l'arbre des identificateurs d'objets utilisé conjointement par l'ISO et le CCITT.

L'appendice I, qui ne fait pas partie de la présente Recommandation, donne des exemples et des directives relatifs à l'utilisation de la notation ASN.1

L'appendice II, qui ne fait pas partie de la présente Recommandation, donne un résumé d'ASN.1 en utilisant la notation du § 5.

Le texte de la présente Recommandation et en particulier les annexes B à D, est agréé conjointement par l'ISO et le CCITT.

## 1 Objectif et domaine d'application

La présente Recommandation spécifie une notation de définition de syntaxe abstraite appelée Notation de syntaxe abstraite numéro un (ASN.1).

La présente Recommandation définit un certain nombre de types simples, avec leurs étiquettes, et spécifie une notation pour la dénotation de ces types et la spécification de valeurs de ces types.

La présente Recommandation définit des mécanismes pour construire de nouveaux types à partir de types plus élémentaires, et spécifie une notation pour définir de tels types structurés et leur affecter des étiquettes, ainsi que pour spécifier des valeurs de ces types.

La présente Recommandation définit des jeux de caractères à utiliser avec ASN.1

La présente Recommandation définit un certain nombre de types utiles (en utilisant ASN.1) auxquels l'utilisateur de l'ASN.1 peut faire référence.

La notation ASN.1 peut être utilisée chaque fois qu'il est nécessaire de définir la syntaxe abstraite d'informations. Elle est en particulier applicable, mais non exclusivement, aux protocoles d'application.

Il est également fait référence à la notation ASN.1 dans d'autres normes et recommandations de la Couche Présentation, qui définissent les règles de codage pour les types simples, les types structurés, les types chaîne de caractères et les types utiles définis dans ASN.1.

## 2 Références

- [1] Recommandation X.200, *Modèle de référence pour l'interconnexion des systèmes ouverts pour les applications du CCITT* (voir aussi la Norme ISO 7498).
- [2] Recommandation X.209, *Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1)* (voir aussi la Norme ISO 8825).
- [3] Recommandation X.216, *Définition du service de présentation pour l'interconnexion des systèmes ouverts pour les applications du CCITT* (voir aussi la Norme ISO 8822).
- [4] Recommandation X.226, *Spécification du protocole de présentation pour l'interconnexion des systèmes ouverts pour les applications du CCITT* (voir aussi la Norme ISO 8823).
- [5] Norme ISO 2014, *Représentation numérique des dates*.
- [6] Norme ISO 2375, *Traitement de l'information – Procédures pour l'enregistrement des séquences d'échappement*.
- [7] Norme ISO 3166, *Code pour la représentation des noms de pays*.
- [8] Norme ISO 3307, *Echange d'informations – Représentation de l'heure*.
- [9] Norme ISO 4031, *Echange d'informations – Représentation des différences d'heure locale*.
- [10] Norme ISO 6523, *Echange de données – Structure d'identification des organisations*.
- [11] Recommandation X.121, *Plan de numérotage international pour les réseaux publics pour données*.

## 3 Définitions

Les définitions de la Recommandation X.200 sont utilisées dans la présente Recommandation.

### 3.1 valeur

Membre distingué d'un ensemble de valeurs.

- 3.2 **type**  
Ensemble de valeurs, doté d'un nom.
- 3.3 **type simple**  
Type défini en spécifiant directement l'ensemble de ses valeurs.
- 3.4 **type structuré**  
Type défini par dénotation d'un ou plusieurs autres types.
- 3.5 **type composant**  
Un des types dénotés dans la définition d'un type structuré.
- 3.6 **étiquette**  
Désignation de type qui est associée à chaque type ASN.1.
- 3.7 **étiquetage**  
Remplacement de l'étiquette existante (éventuellement l'étiquette par défaut) d'un type par une étiquette spécifiée.
- 3.8 **jeu de caractères ASN.1**  
Jeu de caractères, spécifié au Chapitre 7, utilisé dans la notation ASN.1.
- 3.9 **items**  
Séquences de caractères du jeu de caractères ASN.1, dotées d'un nom spécifié au Chapitre 8, qui sont utilisées pour former la notation ASN.1.
- 3.10 **nom de dénotation de type (ou de valeur)**  
Nom associé de façon univoque à un type (ou à une valeur) dans un certain contexte.  
*Remarque* – Des noms de dénotation sont affectés aux types définis dans la présente Recommandation; ils sont universellement disponibles dans ASN.1. D'autres noms de dénotation sont définis dans d'autres normes et recommandations, ils sont uniquement applicables au contexte de la norme ou recommandation concernée.
- 3.11 **règles de codage ASN.1**  
Règles qui spécifient la représentation d'une valeur d'un type quelconque ASN.1, durant le transfert; les règles de codage ASN.1 permettent aux informations transférées d'être identifiées par le destinataire comme une valeur spécifique d'un type ASN.1 particulier.
- 3.12 **type Chaîne de caractères**  
Type dont les valeurs sont des chaînes de caractères d'un certain jeu défini.
- 3.13 **type Booléen**  
Type simple présentant deux valeurs distinguées.
- 3.14 **Vrai**  
Une des valeurs distinguées du type booléen.
- 3.15 **Faux**  
L'autre valeur distinguée du type booléen.
- 3.16 **type Entier**  
Type simple dont les valeurs distinguées sont les nombres entiers positifs et négatifs, y compris zéro (en tant que valeur unique).

*Remarque* - Des règles de codage particulières limitent la plage des valeurs d'un entier, mais ces limitations sont choisies de façon à ne pas affecter un utilisateur d'ASN.1.

### 3.17 **type Enuméré**

Type simple aux valeurs duquel sont attribués des identificateurs distincts, faisant partie de la notation du type.

### 3.18 **type Réel**

Type simple dont les valeurs distinguées (spécifiées au § 16.2) sont des membres de l'ensemble des nombres réels.

### 3.19 **type Chaîne binaire**

Type simple dont chaque valeur distinguée est une séquence ordonnée de zéro, un ou plusieurs bits.

*Remarque* – Les règles de codage ne limitent pas le nombre de bits d'une chaîne binaire.

### 3.20 **type Chaîne d'octets**

Type simple dont chaque valeur distinguée est une séquence ordonnée de zéro, un ou plusieurs octets, chacun étant une séquence ordonnée de 8 bits.

*Remarque* – Les règles de codage ne limitent pas le nombre d'octets d'une chaîne d'octets.

### 3.21 **type Vide**

Type simple présentant une seule valeur, également appelée "vide".

*Remarque* – Le type vide est en général utilisé lorsque plusieurs possibilités sont offertes, mais qu'aucune ne s'applique.

### 3.22 **type Séquence**

Type structuré, défini par dénotation d'une liste ordonnée et fixe de types (dont certains peuvent être déclarés optionnels); chaque valeur du nouveau type est une liste ordonnée de valeurs, une par type composant.

*Remarque* - Lorsqu'un type composant est déclaré optionnel, une valeur du nouveau type ne doit pas nécessairement contenir une valeur de ce type composant.

### 3.23 **type Séquence-de**

Type structuré défini par dénotation d'un seul type existant; chaque valeur du nouveau type est une liste ordonnée de zéro, une ou plusieurs valeurs du type existant.

*Remarque* – Les règles de codage ne limitent pas le nombre de valeurs d'une valeur de séquence-de.

### 3.24 **type Ensemble**

Type structuré, défini par dénotation d'une liste fixe, non ordonnée, de types distincts (dont certains peuvent être déclarés comme optionnels); chaque valeur du nouveau type est une liste non ordonnée de valeurs, une par type composant.

*Remarque* – Quand un type composant est déclaré optionnel, le nouveau type ne doit pas nécessairement contenir une valeur de ce type composant.

### 3.25 **type Ensemble-de**

Type structuré, défini par dénotation d'un seul type existant; chaque valeur du nouveau type est une liste non ordonnée de zéro, une ou plusieurs valeurs du type existant.

*Remarque* – Les règles de codage ne limitent pas le nombre de valeurs d'un ensemble-de.

### 3.26 **type Etiqueté**

Type défini par dénotation d'un type existant simple et d'une étiquette; le nouveau type est isomorphe au type existant, mais il en est distinct.

### 3.27 **type Choix**

Type structuré, défini par dénotation d'une liste fixe, non ordonnée, de types distincts; chaque valeur du nouveau type est une valeur d'un des types composants.

### 3.28 **type Sélection**

Type structuré, défini par dénotation d'un type composant d'un type choix.

### 3.29 **type Quelconque**

Type choix dont les types composants ne sont pas spécifiés, mais restreints à l'ensemble des types qui peuvent être définis en utilisant ASN.1.

### 3.30 **type Externe**

Type dont les valeurs distinguées ne peuvent pas être déduites de leur caractéristique d'externe, mais qui peuvent être déduites de leur codage; les valeurs peuvent être décrites, mais pas nécessairement, en utilisant ASN.1; de même, leur codage peut, mais pas nécessairement, être conforme aux règles de codage d'ASN.1.

### 3.31 **objet informationnel**

Information, définition ou spécification bien définies, qui nécessitent un nom pour être identifiées dans une communication.

### 3.32 **identificateur d'objet**

Valeur (distinguable de toutes les autres valeurs), associée à un objet informationnel.

### 3.33 **type Identificateur d'objet**

Type dont les valeurs distinguées sont l'ensemble de tous les identificateurs d'objet affectés conformément aux règles de la présente Recommandation.

*Remarque* – Les règles de la présente Recommandation permettent à des autorités très diverses, d'associer en toute indépendance des identificateurs d'objets à des objets informationnels.

### 3.34 **type Descripteur d'objet**

Type dont chaque valeur distinguée est un texte lisible par l'homme, donnant une brève description d'un objet informationnel.

*Remarque* – Une valeur de descripteur d'objet est généralement, mais pas toujours, associée à un objet informationnel unique. Seule une valeur d'identificateur d'objet identifie de manière non ambiguë un objet informationnel.

### 3.35 **définitions récursives**

Ensemble de définitions ASN.1 qui ne peuvent pas être réordonnées de telle sorte que tous les types utilisés dans une construction soient définis avant la définition de cette construction.

*Remarque* – Les définitions récursives sont autorisées en ASN.1: il appartient à l'utilisateur de s'assurer que les valeurs (des types résultants) utilisées, ont une représentation finie.

### 3.36 **module**

Une ou plusieurs définitions de type ou de valeur utilisant la notation ASN.1, et encadrées en utilisant la notation d'un module ASN.1 (voir le § 9).

### 3.37 **production**

Partie de la notation formelle utilisée pour spécifier ASN.1, dans laquelle des séquences autorisées d'items sont associées à un nom qui peut être employé pour dénoter ces séquences dans la définition de nouveaux ensembles de séquences autorisées.

### 3.38 **Temps Universel Coordonné (UTC)**

Mesure du temps conservée par le Bureau international de l'heure, et servant de base à la diffusion coordonnée des fréquences standard et des signaux horaires.

*Remarque 1* – L'origine de cette définition est la Recommandation 460-2 du Comité consultatif international des radiocommunications (le CCIR). Le CCIR a également défini le sigle du temps universel coordonné: UTC.

*Remarque 2* – L'UTC est également appelé heure GMT et des signaux horaires appropriés sont diffusés régulièrement.

### 3.39 **utilisateur (d'ASN.1)**

Personne physique ou morale qui définit la syntaxe abstraite d'une information particulière en utilisant ASN.1.

### 3.40 **sous-type (d'un type parent)**

Type dont les valeurs sont spécifiées comme sous-ensemble des valeurs d'un autre type (le type parent).

### 3.41 **type parent (d'un sous-type)**

Type utilisé pour définir un sous-type.

*Remarque* – Le type parent peut lui-même être un sous-type d'un autre type.

### 3.42 **spécification d'un sous-type**

Notation qui peut être utilisée, associée à la notation d'un type, pour définir un sous-type de ce type.

### 3.43 **ensemble des valeurs d'un sous-type**

Notation faisant partie de la spécification d'un sous-type, et spécifiant un ensemble des valeurs du type parent qui doivent être incluses dans le sous-type.

3.44 La présente Recommandation utilise les termes suivants définis dans la Recommandation X.216:

- a) valeur de données de présentation;
- b) (une) syntaxe abstraite;
- c) nom de syntaxe abstraite;
- d) nom de syntaxe de transfert.

3.45 La présente Recommandation utilise également les termes suivants définis dans la Norme ISO 6523:

- a) organisation originaire;
- b) code d'organisation;
- c) prescripteur de code international.

3.46 La présente Recommandation utilise le terme suivant défini dans la Recommandation X.226:

- a) identificateur de contexte de présentation.

## 4 **Abréviations**

ASN.1	Notation de syntaxe abstraite numéro un.
UTC	Temps universel coordonné.
ICD	Prescripteur de code international.
IPD	Indicatif de pays pour la transmission de données.
CIRD	Code d'identification de réseau pour données.

## 5 Notation utilisée dans la présente Recommandation

La notation ASN.1 consiste en une séquence de caractères du jeu de caractères ASN.1, spécifiée au § 7.

Chaque utilisation de la notation ASN.1 contient des caractères du jeu ASN.1, regroupés en items. Le § 8 spécifie toutes les séquences de caractères formant des items ASN.1, avec leur nom.

La notation ASN.1 est définie au § 9 (et suivants), en spécifiant l'ensemble des séquences d'items qui forment des occurrences "valides" de la notation ASN.1, avec la sémantique de chaque séquence.

Pour spécifier ces ensembles, la présente Recommandation utilise une notation formelle définie dans les paragraphes suivants.

### 5.1 Productions

Un nouvel ensemble (plus complexe) de séquences ASN.1 est défini au moyen d'une production. Une production utilise les noms des ensembles de séquences définis dans la présente Recommandation pour former un nouvel ensemble de séquences, en spécifiant:

- a) soit que le nouvel ensemble de séquences doit être composé de n'importe quelle séquence contenue dans l'un quelconque des ensembles d'origine;
- b) soit que le nouvel ensemble doit être composé d'une séquence qui peut être générée en prenant une séquence et une seule dans chaque ensemble et en les juxtaposant dans un ordre spécifié.

Chaque production comprend les parties suivantes, sur une ou plusieurs lignes, dans l'ordre:

- a) le nom du nouvel ensemble de séquences;
- b) les caractères:

::=

- c) un ou plusieurs ensembles possibles de séquences, tel que défini au § 5.2, séparés par le caractère:

|

Une séquence figure dans le nouvel ensemble si elle figure dans un ou plusieurs des ensembles possibles. Le nouvel ensemble est dénoté dans la présente Recommandation par le nom, mentionné en a) ci-dessus.

*Remarque* – Si la même séquence figure dans plusieurs ensembles de séquences de la partie droite de la production, toute ambiguïté sémantique de la notation résultante est résolue par d'autres parties de la séquence ASN.1 complète.

### 5.2 Partie droite d'une production

Chacun des ensembles possibles de séquences de la partie droite d'une production est spécifié par une liste de noms. Chaque nom est soit le nom d'un item, soit le nom d'un ensemble de séquences défini par une production dans la présente Recommandation.

L'ensemble des séquences défini dans la partie droite d'une production comprend toutes les séquences obtenues en prenant l'une des séquences (ou des items) associé(e)s au premier nom, en combinaison avec (et suivi(e)s de) n'importe laquelle des séquences (ou items associé(e)s au second nom, combiné(e)s avec (et suivi(e)s de) n'importe laquelle des séquences (ou items) associé(e)s au troisième nom, et ainsi de suite jusque et y compris le dernier nom (ou item).

### 5.3 Exemple de production

Valeur-Chaîne-Binaire ::=  
chaîne-b |  
chaîne-h |  
{Liste-Identificateurs}

est une production qui associe au nom **Valeur-Chaîne-Binaire** les séquences suivantes:

- a) toute chaîne-b (un item);
- b) toute chaîne-h (un item);
- c) toute séquence associée à **Liste-Identificateur**, précédée d'une { et suivie d'une }.

*Remarque* – { et } sont les noms des items contenant respectivement les seuls caractères { et } (voir le § 8).

Dans cet exemple, **Liste-Identificateurs** serait défini par une production, avant ou après la production définissant **Valeur-Chaîne-Binaire**.

#### 5.4 Disposition

Chaque production utilisée dans la présente Recommandation est précédée et suivie d'une ligne vide. Il ne figure pas de ligne vide à l'intérieur des productions. Les productions peuvent être sur une seule ligne ou plusieurs. La disposition n'est pas significative.

#### 5.5 Récursivité

Les productions de la présente Recommandation sont souvent récursives. Dans ce cas, les productions doivent être réappliquées de façon répétitive, jusqu'au moment où aucune nouvelle séquence n'est générée.

*Remarque* – Dans de nombreux cas, cette réapplication produit un ensemble non limité de séquences autorisées, dont certaines sont elles-mêmes non limitées. Ce n'est pas une erreur.

#### 5.6 Dénotation d'un ensemble de séquences

La présente Recommandation dénote un ensemble de séquences (faisant partie de la notation ASN.1) en citant le nom de la partie gauche d'une production (avant le ::=); le nom est mis entre guillemets (") pour le distinguer du texte en langage naturel sauf s'il apparaît comme partie d'une production.

#### 5.7 Références à un item

La présente Recommandation dénote un item en citant son nom; le nom est mis entre guillemets (") pour le distinguer du texte en langage naturel sauf s'il apparaît comme partie d'une production.

TABLEAU 1/X.208

#### Affectations des étiquettes de la classe "UNIVERSAL"

UNIVERSAL 1	Type Booléen
UNIVERSAL 2	Type Entier
UNIVERSAL 3	Type Chaîne binaire
UNIVERSAL 4	Type Chaîne d'octets
UNIVERSAL 5	Type Vide
UNIVERSAL 6	Type Identificateur d'objet
UNIVERSAL 7	Type Descripteur d'objet
UNIVERSAL 8	Type Externe
UNIVERSAL 9	Type Réel
UNIVERSAL 10	Type Enuméré
UNIVERSAL 12 à 15	Réservées pour des versions ultérieures de la présente Recommandation
UNIVERSAL 16	Types Séquence et Séquence-de
UNIVERSAL 17	Types Ensemble et Ensemble-de
UNIVERSAL 18 à 22, 25 à 27	Types Chaîne de caractères
UNIVERSAL 23, 24	Types Heure
UNIVERSAL 28 à . . .	Réservées pour des versions ultérieures de la présente Recommandation

## 5.8 *Étiquettes*

Une étiquette est spécifiée en donnant sa classe et son numéro dans cette classe. La classe est l'une des suivantes:

universal  
application  
private  
context-specific.

Le numéro est un entier non négatif, spécifié en notation décimale.

Les limitations applicables aux étiquettes par l'utilisateur d'ASN.1 sont spécifiées au § 26.

Les étiquettes de la classe "universal" sont affectées de telle sorte que pour les types structurés, la structure du niveau le plus élevé peut se déduire de l'étiquette, alors que pour les types simples, le type peut se déduire de l'étiquette. Le tableau 1/X.208 résume les affectations des étiquettes de la classe "universal", qui sont spécifiées dans la présente Recommandation.

*Remarque* – Des étiquettes supplémentaires de la classe "universal" sont réservées pour affectation par des versions ultérieures de la présente Recommandation.

## 6 **Utilisation de la notation ASN.1**

6.1 La notation ASN.1 d'une définition de type est "Type" (voir le § 12.1).

6.2 La notation ASN.1 de la valeur d'un type est "Valeur" (voir le § 12.7).

*Remarque* – Il n'est en général pas possible d'interpréter la notation d'une valeur sans en connaître le type.

6.3 La notation ASN.1 d'affectation d'un type à un nom de dénotation de type est "Affectation-type" (voir le § 11).

6.4 La notation ASN.1 d'affectation d'une valeur à un nom de dénotation de valeur est "Affectation-valeur" (voir le § 11.2).

6.5 Les notations "Affectation-type" et "Affectation-valeur" doivent être uniquement utilisées dans la notation "Définition-Module" (mais voir le § 9.1).

## SECTION 1 – SPÉCIFICATION DE LA NOTATION ASN.1

## 7 **Le jeu de caractères ASN.1**

7.1 Un élément ASN.1 consiste en une séquence des caractères indiqués dans le tableau 2/X.208, sous réserve de ce qui est spécifié aux § 7.2 et § 7.3.

TABLEAU 2/X.208

### Caractères ASN.1

A à Z
a à z
0 à 9
:=, { } < .
() [ ] - ' "

*Remarque 1* – Les caractères supplémentaires > et | sont utilisés dans la notation des macros (voir annexe A).

*Remarque 2* – Quand des normes dérivées équivalentes sont élaborées par des Comités de normalisation nationaux, des caractères supplémentaires peuvent apparaître dans les items suivants (dont les cinq derniers sont définis dans l'annexe A):

dénotation-type	(§ 8.2.1)
identificateur	(§ 8.3)
dénotation-valeur	(§ 8.4)
dénotation-module	(§ 8.5)
dénotation-macro	(§ A.2.1)
dénotation-production	(§ A.2.2)
dénotation-type-local	(§ A.2.3)
dénotation-valeur-locale	(§ A.2.4)
chaîne-a	(§ A.2.7)

*Remarque 3* – Quand des caractères supplémentaires sont introduits pour pouvoir utiliser un langage dans lequel la distinction entre les majuscules et les minuscules n'a pas de signification, la distinction syntaxique réalisée en imposant que le premier caractère de certains des items ASN.1 ci-dessus soit une lettre majuscule ou minuscule, doit être obtenue d'une autre façon.

7.2 Lorsque la notation est utilisée pour spécifier la valeur d'un type chaîne de caractères, tous les caractères du jeu de caractères défini peuvent apparaître dans la notation ASN.1, encadrés des caractères " " (voir le § 8.11).

7.3 Des caractères supplémentaires peuvent apparaître dans l'item "commentaire" (voir le § 8.6).

7.4 Aucune signification particulière n'est accordée à la typographie, à la taille, à la couleur, à la graisse ou aux autres caractéristiques d'affichage ou d'impression.

7.5 Les lettres majuscules et minuscules sont considérées comme distinctes.

## **8 Items ASN.1**

### *8.1 Règles générales*

8.1.1 Les paragraphes suivants spécifient les caractères des items ASN.1. Dans chaque cas, le nom de l'item est donné avec la définition des séquences de caractères qui le forment.

*Remarque* – L'annexe A spécifie les items supplémentaires utilisés dans la notation des macros.

8.1.2 Chaque item spécifié dans les paragraphes suivants apparaît sur une seule ligne et ne contient pas d'espace (sauf l'item "commentaire").

8.1.3 La longueur d'une ligne n'est pas limitée.

8.1.4 Les items des séquences spécifiées par la présente Recommandation (la notation ASN.1) peuvent apparaître sur une ou plusieurs lignes et être séparées par un ou plusieurs espaces ou lignes vides.

8.1.5 Un item est séparé du suivant par un espace, ou placé sur une ligne différente, si le (ou les) caractère(s) initiaux de l'item suivant sont un (ou des) caractère(s) dont l'inclusion à la fin des caractères du premier item est autorisée.

### *8.2 Dénotations de types*

Nom de l'item: dénotation-type

8.2.1 Une "dénotation-type" consiste en un nombre quelconque (un ou plusieurs) de lettres, chiffres et traits d'union. Le premier caractère doit être une majuscule. Le dernier caractère ne peut pas être un trait d'union. Un trait d'union ne doit pas être immédiatement suivi par un autre trait d'union.

*Remarque* – Les règles concernant le trait d'union sont destinées à éviter tout risque de confusion avec un commentaire (éventuellement placé à la suite).

8.2.2 Une "dénotation-type" ne doit pas être une des séquences de caractères réservées figurant au tableau 3/X.208.

*Remarque* – Le § A.2.9 spécifie d'autres séquences de caractères réservées lors d'une définition de macros.

### 8.3 *Identificateurs*

Nom de l'item: identificateur

Un "identificateur" consiste en un nombre quelconque (un ou plusieurs) de lettres, chiffres et traits d'union. Le premier caractère doit être une minuscule. Le dernier caractère ne peut pas être un trait d'union. Un trait d'union ne peut pas être immédiatement suivi par un autre trait d'union.

*Remarque* – Les règles concernant le trait d'union sont destinées à éviter tout risque de confusion avec un commentaire (placé éventuellement à la suite).

TABLEAU 3/X.208

#### Séquences de caractères réservées

BOOLEAN	BEGIN
INTEGER	END
BIT	DEFINITIONS
STRING	EXPLICIT
OCTET	ENUMERATED
NULL	EXPORTS
SEQUENCE	IMPORTS
OF	REAL
SET	INCLUDES
IMPLICIT	MIN
CHOICE	MAX
ANY	SIZE
EXTERNAL	FROM
OBJECT	WITH
IDENTIFIER	COMPONENT
OPTIONAL	PRESENT
DEFAULT	ABSENT
COMPONENTS	DEFINED
UNIVERSAL	BY
APPLICATION	PLUS-INFINITY
PRIVATE	MINUS-INFINITY
TRUE	TAGS
FALSE	

### 8.4 *Dénotations de valeur*

Nom de l'item: dénotation-valeur

Une "dénotation-valeur" est composée de la séquence de caractères spécifiée pour un "identificateur" au § 8.3. Lors de l'analyse d'une occurrence de cette notation, une "dénotation-valeur" se distingue d'un "identificateur" par le contexte dans lequel elle apparaît.

### 8.5 *Dénotations d'un module*

Nom de l'item: dénotation-module

Une "dénotation-module" est composée de la séquence de caractères spécifiée pour une "dénotation-type" au § 8.2. Lors de l'analyse d'une occurrence de cette notation, une "dénotation-module" se distingue d'une "dénotation-type" par le contexte dans lequel elle apparaît.

## 8.6 *Commentaire*

Nom du commentaire: commentaire

8.6.1 Un "commentaire" n'a pas de dénotation dans la définition de la notation ASN.1: il peut apparaître n'importe où entre d'autres items ASN.1, et n'a pas de signification.

8.6.2 Un "commentaire" commence par deux traits d'union consécutifs et se termine au premier couple de traits d'union consécutifs rencontré ou à la fin de la ligne. Un commentaire ne doit pas contenir de couples de traits d'union consécutifs autres que ceux par lequel il commence et éventuellement se termine. Il peut inclure des caractères qui n'appartiennent pas au jeu spécifié au § 7.1 (voir le § 7.3).

## 8.7 *Item vide*

Nom de l'item: vide

L'item "vide" ne contient aucun caractère. Il est utilisé dans la notation du § 5, quand plusieurs ensembles possibles de séquences sont spécifiés, pour indiquer qu'il peut y avoir absence de toute possibilité.

## 8.8 *Item nombre*

Nom de l'item: nombre

Un "nombre" comprend un ou plusieurs chiffres. Le premier chiffre doit être différent de zéro, sauf si le "nombre" comprend un seul chiffre.

## 8.9 *Item chaîne binaire*

Nom de l'item: chaîne-b

Une "chaîne-b" comprend un nombre quelconque (éventuellement zéro) de zéros et de uns, précédés d'un caractère ' unique et suivis du couple de caractères:

'B

Exemple: '01101100'B

## 8.10 *Item chaîne hexadécimale*

Nom de l'item: chaîne-h.

8.10.1 Une "chaîne-h" est composée d'un nombre quelconque (éventuellement zéro) de caractères, choisis parmi:

A B C D E F 0 1 2 3 4 5 6 7 8 9

précédés d'un caractère ' unique et suivis du couple de caractères:

'H

Exemple: 'AB0196'H

8.10.2 Chaque caractère indique la valeur d'un semi-octet, exprimé en représentation hexadécimale.

## 8.11 *Item chaîne de caractères*

Nom de l'item: chaîne-c

Une "chaîne-c" est composée d'un nombre quelconque (éventuellement zéro) de caractères du jeu dénoté par un type chaîne de caractères, précédés et suivis de “. Si le jeu de caractères comprend le caractère “, il est représenté dans la "chaîne-c" par un couple de “. Le jeu de caractères concerné ne se limite pas au jeu du tableau 2/X.208, il est déterminé par le type pour lequel la "chaîne-c" est une valeur (voir le § 7.2).

Exemple: "宛.姐.虻.鉛.拘"

8.12 *Item affectation*

Nom de l'item: "::~="

Cet item est composé de la séquence de caractères:

::=

*Remarque* – Cette séquence ne doit contenir aucun caractère espace (voir le § 8.1.2).

8.13 *Items caractère unique*

Noms des items:

{  
}  
<  
,  
.  
(  
)  
[  
]  
- (trait d'union)  
;

Un item dont le nom est l'un de ceux indiqués ci-dessus est composé du seul caractère formant ce nom.

8.14 *Items mot clé*

Noms des items:

BOOLEAN  
INTEGER  
BIT  
STRING  
OCTET  
NULL  
SEQUENCE  
OF  
SET  
IMPLICIT  
CHOICE  
ANY  
EXTERNAL  
OBJECT  
IDENTIFIER  
OPTIONAL  
DEFAULT  
COMPONENTS  
UNIVERSAL  
APPLICATION  
PRIVATE  
TRUE  
FALSE  
BEGIN  
END  
DEFINITIONS  
EXPLICIT  
ENUMERATED  
EXPORTS  
IMPORTS  
REAL  
INCLUDES  
MIN  
MAX

SIZE  
FROM  
WITH  
COMPONENT  
PRESENT  
ABSENT  
DEFINED  
BY  
PLUS-INFINITY  
MINUS-INFINITY  
TAGS

Les items dont les noms sont donnés ci-dessus sont composés de la séquence des caractères du nom.

*Remarque 1* – Aucun espace ne doit figurer dans ces séquences.

*Remarque 2* – Les séquences ci-dessus qui sont différentes des séquences réservées indiquées au § 8.2.2, sont distinguées des autres items contenant les mêmes caractères par le contexte dans lequel elles figurent.

## 9 Définition d'un module

9.1 Une "Définition-Module" est spécifiée par les productions suivantes:

Définition-Module ::=

Identificateur-Module  
DEFINITIONS  
Défaut-Etiquette  
"::="

BEGIN

Corps-Module

END

Défaut-Etiquette ::=

EXPLICIT TAGS |  
IMPLICIT TAGS |  
vide

Identificateur-Module ::=

Dénotation-Module  
Identificateur-Affecté

Identificateur-Affecté ::=

Valeur-Identificateur-Objet |  
vide

Corps-Module ::=

Exportations Importations Liste-Affectations |  
vide

Exportations ::=

EXPORTS Symboles-Exportés; |  
vide

Symboles-Exportés ::=

Liste-Symboles |  
vide

Importations ::=

IMPORTS Symboles-Importés; |  
vide

Symboles-Importés ::=  
 Symboles-De-Liste-Modules |  
 vide

Symboles-De-Liste-Modules ::=  
 Symboles-De-Module Symboles-De-Liste-Modules |  
 Symboles-De-Module

Symboles-De-Module ::=  
 Liste-Symboles FROM Identificateur-Module

Liste-Symboles ::= Symbole, Liste-Symboles | Symbole

Symbole ::= dénotation-type | dénotation-valeur

Liste-Affectations ::=  
 Affectation Liste-Affectations |  
 Affectation

Affectation ::=  
 Affectation-Type | Affectation-Valeur

*Remarque 1* – L'annexe A spécifie une séquence "Définition-Macros" qui peut également figurer dans la "Liste-Affectations". Les notations spécifiées par une définition de macro peuvent apparaître dans le même module, avant ou après cette définition.

*Remarque 2* – Dans certains cas particuliers (et déconseillés) pour des exemples ou pour la définition de types d'étiquettes de classe "universal", le "Corps-Module" peut être utilisé à l'extérieur d'une "Définition-Module".

*Remarque 3* – Les productions "Affectation-type" et "Affectation-valeur" sont spécifiées au § 11.

*Remarque 4* – Le regroupement de types de données ASN.1 en module ne détermine pas nécessairement la formation de valeurs de données de présentation, dans les syntaxes abstraites mentionnées, pour la définition de contextes de présentation.

*Remarque 5* – La valeur de "Etiquette-Défaut" de la définition d'un module, affecte uniquement les types définis explicitement dans ce module. Elles n'affectent pas l'interprétation des types importés.

*Remarque 6* – Une "Dénotation-Macro" (voir annexe A) peut également apparaître comme un "Symbole".

9.2 Si "Etiquette-Défaut" est vide, elle est prise comme étant "EXPLICIT TAGS".

*Remarque* – Le § 26 donne la signification de "EXPLICIT TAGS" et "IMPLICIT TAGS".

9.3 La "Dénotation-Module" apparaissant dans la production "Définition-Module" est appelée **nom du module**. Les noms des modules sont choisis en sorte de garantir la cohérence et le caractère complet de toutes les séquences "Affectation" apparaissant dans le "Corps-Module" de toutes les séquences "Définition-Module" de ce nom. Un ensemble de séquences "Affectation" est cohérent et complet si, pour chaque "Dénotation-Type" ou "Dénotation-Valeur" qui y figure, il existe exactement une "Affectation-type" ou une "Affectation-valeur" associant (respectivement) le nom à un type ou une valeur, ou exactement un "Symboles-De-Module" dans lequel la "Dénotation-type" ou "Dénotation-valeur" apparaît comme un "Symbole".

9.4 Les noms de module ne doivent être utilisés qu'une seule fois (sauf dans les cas spécifiés dans le § 9.10) dans le domaine d'intérêt de la définition du module.

*Remarque* – Il est recommandé de donner aux modules définis dans les Normes ISO un nom de module de la forme:

ISOxxxx-yyyy

où xxxx est le numéro de la Norme internationale et yyyy est un sigle approprié à cette Norme internationale (par exemple, JTM, FTAM ou CCR). Une convention similaire peut être appliquée par d'autres organismes de normalisation.

9.5 Si "Identificateur-Affecté" comprend une "Valeur-Identificateur-Objet", elle identifie de façon non ambiguë et unique le module.

*Remarque* – Il est recommandé d'affecter un identificateur d'objet de manière telle que les autres puissent se référer sans ambiguïté au module.

9.6 La séquence "Identificateur-Module" de "Symboles-De-Module" doit figurer dans la "Définition-Module" d'un autre module: la "dénotation-module" peut être différente dans les deux cas, sauf si elle comprend une "Valeur-Identificateur-Module".

*Remarque 1* – Une "Dénotation-module" différente de celle utilisée dans l'autre module ne doit être utilisée que lorsque les symboles doivent être importés de deux modules de même nom (les modules étant désignés sans tenir compte du § 9.4). L'utilisation d'autres noms distincts rend ces noms disponibles pour une utilisation dans le corps du module (voir le § 9.8).

*Remarque 2* – Quand une "Dénotation-module" et une "Valeur-Identificateur-Objet" sont utilisées dans la dénotation d'un module, cette dernière est considérée comme définitive.

9.7 Chaque "Symboles" de "Symboles-Exportés" doit être défini dans le module en cours de construction.

*Remarque* – Il est recommandé d'inclure dans les "Symboles-Exportés" chaque symbole dont la dénotation à partir de l'extérieur du module est appropriée. S'il n'existe pas de tels symboles, il faut choisir l'alternative "vide" de "Symboles-Exportés" (et non de "Exports").

9.8 Chaque "Symbole" des "Symboles-De-Module" doit être défini dans le module désigné par "Identificateur-Module" de "Symboles-De-Module". Si "Exports" est utilisé dans la définition de ce module, le "Symbole" doit figurer dans ces "Symboles-Exportés".

9.9 Un "Symbole" des "Symboles-De-Module" peut figurer dans un "Corps-Module" d'un "Type-Défini" (si c'est une "dénotation-type") ou une "Valeur-Défini" (si c'est une "dénotation-valeur"). La signification associée au "Symboles" est celle qu'il a dans le module désigné par l'"Identificateur-Module" correspondant. Quand le "Symbole" figure également dans une "Liste-Affectations" (déconseillé) ou figure dans une ou plusieurs occurrences de "Symboles-De-Module", il doit être uniquement utilisé dans une "Dénotation-Type-Externe" ou une "Dénotation-Valeur-Externe", dont la "dénotation-module" est celle figurant dans "Symboles-De-Module" (voir le § 9.10). S'il ne figure pas, il peut être utilisé directement dans un "Type-Défini" ou une "Valeur-Défini".

9.10 Sauf dans les cas spécifiés au § 9.9, une "dénotation-type" ou "dénotation-valeur" doit être issue d'un module différent de celui dans lequel elle est définie à l'aide d'une "Dénotation-type-externe" ou "Dénotation-valeur-externe", spécifiée par les productions suivantes:

Dénotation-Type-Externe ::=  
    dénotation-module  
    dénotation-type

Dénotation-valeur-externe ::=  
    dénotation-module  
    dénotation-valeur

## 10 Dénotation des définitions de types et de valeurs

### 10.1 Les productions

Type-Défini ::=  
    Dénotation-type-externe |  
    dénotation-type

Valeur-Définie ::=  
    Dénotation-valeur-externe |  
    dénotation-valeur

spécifient les séquences qui seront utilisées pour les dénnotations des définitions de types et de valeurs.

10.2 En-dehors des cas spécifiés au § 9.10, les possibilités "dénotation-type" et "dénotation-valeur" ne devront pas être utilisées si la dénotation provient du module dans lequel un type ou une valeur est affectée (voir les § 11.1 et 11.2) à la "dénotation-type" ou "dénotation-valeur".

- 10.3 Les "Dénotation-type-externe" et "Dénotation-valeur-externe" ne seront utilisées que si un type ou une valeur a été respectivement affectée à la "dénotation-type" ou "dénotation-valeur" correspondante (voir les § 11.1 et 11.2) dans la "dénotation-module" correspondante.

## 11 Affectation de types et de valeurs

- 11.1 Une "dénotation-type" est affectée à un type par la notation spécifiée dans la production "Affectation-type":

```
Affectation-type ::= dénotation-type
                  "::~"
                  Type
```

La "dénotation-type" ne doit pas être l'un des noms utilisés pour dénoter les types de chaînes de caractères définis dans la section 2, ni l'un des noms utilisés pour dénoter les types définis dans la section 3.

- 11.2 Une "dénotation-valeur" est affectée à une valeur par la notation spécifiée par la production "Affectation-valeur":

```
Affectation-valeur ::= dénotation-valeur
                    Type
                    "::~"
                    Valeur
```

La "Valeur" affectée à la "dénotation-valeur" doit être une notation valide (voir le § 12.7) d'une valeur du type défini par "Type".

## 12 Définition des types et valeurs

- 12.1 Un type est dénoté par l'une des séquences "Type":

```
Type ::= Type-Prédéfini | Type-Défini | Sous-type
        (voir le § 10.1) (voir le § 37)
```

Type-Prédéfini ::=

```
Type-Booléen |
Type-Entier |
Type-Chaîne-Binaire |
Type-Chaîne-Octets |
Type-Vide |
Type-Séquence |
Type-Séquence-De |
Type-Ensemble |
Type-Ensemble-De |
Type-Choix |
Type-Sélection |
Type-Etiqueté |
Type-Quelconque |
Type-Identificateur-Objet |
Type-Chaîne-Caractères |
Type-Utile |
Type-Énuméré |
Type-Réel |
```

*Remarque 1* – Une notation de type défini dans une macro peut également être utilisée comme séquence pour "Type" (voir annexe A).

*Remarque 2* – Des types prédéfinis supplémentaires pourront être définis dans les versions ultérieures de la présente Recommandation.

- 12.2 La notation "Type-Prédéfini" est spécifiée dans les paragraphes suivants.

12.3 La notation "Sous-type" est spécifiée dans la section 4.

12.4 Le type dénoté est le type défini par le "Type-Prédéfini" ou le "Sous-type" affecté au "Type-Défini".

12.5 Le type peut être nommé dans certaines notations dans lesquelles il est dénoté. Dans ces cas, la présente Recommandation spécifie l'utilisation de la notation "Type-Nommé":

Type-Nommé ::=

Type-Identificateur |  
Type |  
Type-Sélection

La notation "Type-Sélection" et la notation de la valeur correspondante sont spécifiées au § 25.

Remarque – La notation "Type-Sélection" contient un "identificateur" qui peut faire partie de la notation de valeur quand le "Type-Sélection" est utilisé comme "Type-Nommé" (voir le § 25.1).

12.6 L'"identificateur" ne fait pas partie du type et n'a pas d'effet sur celui-ci. Le type dénoté par une séquence "Type-Nommé" est celui dénoté par la séquence "Type" contenue.

12.7 La valeur d'un type est spécifiée par l'une des séquences "Valeur":

Valeur ::= Valeur-Prédéfini | Valeur-Défini

Valeur-Prédéfini ::=

Valeur-Booléen |  
Valeur-Entier |  
Valeur-Chaîne-Binaire |  
Valeur-Chaîne-Octets |  
Valeur-Vide |  
Valeur-Séquence |  
Valeur-Séquence-De |  
Valeur-Ensemble |  
Valeur-Ensemble-De |  
Valeur-Choix |  
Valeur-Sélection |  
Valeur-Etiqueté |  
Valeur-Quelconque |  
Valeur-Identificateur-Objet |  
Valeur-Chaîne-Caractères |  
Valeur-Enuméré |  
Valeur-Réel |

Remarque – Une notation de valeur définie dans une macro peut également être utilisée comme séquence pour "Valeur" (voir annexe A).

12.8 Si le type est défini à l'aide d'une des notations de la colonne de gauche ci-après, la valeur doit être spécifiée en utilisant la notation de droite correspondante:

<b>Notation du type</b>	<b>Notation de la valeur</b>
Type-Booléen	Valeur-Booléen
Type-Entier	Valeur-Entier
Type-Chaîne-Binaire	Valeur-Chaîne-Binaire
Type-Chaîne-Octets	Valeur-Chaîne-Octets
Type-vide	Valeur-Vide
Type-Séquence	Valeur-Séquence
Type-Séquence-De	Valeur-Séquence-De
Type-Ensemble	Valeur-Ensemble
Type-Ensemble-De	Valeur-Ensemble-De
Type-Choix	Valeur-Choix

Type-Etiqueté	Valeur-Etiqueté
Type-Quelconque	Valeur-Quelconque
Type-Identificateur-Objet	Valeur-Identificateur-Objet
Type-Chaîne-Caractères	Valeur-Chaîne-Caractères
Type-Énuméré	Valeur-Énuméré
Type-Réel	Valeur-Réel

*Remarque* – Des notations de valeur additionnelles pourront être définies dans des versions ultérieures de la présente Recommandation.

La notation de la valeur du type "Type-Défini" doit être celle du type utilisé dans la production de ce "Type-Défini".

12.9 La notation de la valeur d'un type défini par la notation "Type-Utile" est spécifiée dans la section 3.

12.10 La notation "Valeur-Prédéfini" est spécifiée dans les paragraphes suivants.

12.11 La valeur d'un type dénoté à l'aide de la notation "Type-Nommé" doit être définie par la notation "Valeur-Nommé"

Valeur-Nommé ::=  
           Valeur-Identificateur |  
           Valeur

où "identificateur" (le cas échéant) est le même que celui utilisé dans la notation "Type-Nommé". Le § 25.2 spécifie d'autres limitations applicables à la "Valeur-Nommé" lorsque le "Type-Nommé" était un "Type-Sélection".

*Remarque* – "identificateur" fait partie de la notation et non de la valeur proprement dite.

12.12 "identificateur" figure dans la "Valeur-Nommé" si et seulement s'il figurait dans le "Type-Nommé".

*Remarque* – Un "identificateur" figure toujours dans le cas d'un "Type-Sélection".

### 13 Notation du type Booléen

13.1 Le type Booléen (voir le § 3.13) est dénoté par la notation "Type-Booléen":

Type-Booléen ::= BOOLEAN

13.2 L'étiquette des types définis par cette notation est de classe "universal", numéro 1.

13.3 La valeur d'un type Booléen (voir les § 3.14 et 3.15) est définie par la notation "Valeur-Booléen":

Valeur-Booléen ::= TRUE | FALSE

### 14 Notation du type Entier

14.1 Le type Entier (voir le § 3.16) est dénoté par la notation "Type-Entier":

Type-Entier ::=  
           INTEGER |  
           INTEGER {Liste-Nombres-Nommés}

Liste-Nombres-Nommés ::=  
           Nombres-Nommés |  
           Liste-Nombres-Nommés, Nombres-Nommés

Nombres-Nommés ::=  
           Identificateur (Nombre-Signé) |  
           Identificateur (Valeur-Défini)

Nombre-Signé ::= nombre | -nombre

- 14.2 La seconde alternative de "Nombre-Signé" ne sera pas utilisée si le "nombre" est zéro.
- 14.3 Dans la définition d'un type, la "Liste-Nombres-Nommés" n'est pas significative. Elle est seulement utilisée dans la notation spécifiée au § 14.9.
- 14.4 La "Valeur-Défini" sera une dénotation d'une valeur de type entier ou d'un type dérivé d'un entier par étiquetage ou par sous-type.
- 14.5 Les valeurs de chaque "Nombre-Signé" ou "Valeur-Défini" apparaissant dans la "Liste-Nombres-Nommés" doivent être différentes et représenter une valeur du type entier.
- 14.6 Chaque "identificateur" apparaissant dans la "Liste-Nombre-Nommés" doit être différent.
- 14.7 L'ordre des séquences "Nombres-Nommés" "de la Liste-Nombre-Nommés" n'a pas d'importance.
- 14.8 L'étiquette des types définis par cette notation est de classe "universal", numéro 2.
- 14.9 La valeur d'un type Entier est définie par la notation "Valeur-Entier":

```
Valeur-Entier ::=
    Nombre-Signé |
    Identificateur
```

14.10 L'item "identificateur" de "Valeur-Entier" doit être égal à l'item "identificateur" de la séquence "Type-Entier" à laquelle la valeur est associée, et doit représenter le nombre correspondant.

*Remarque* - Pour définir une valeur d'Entier pour laquelle un "identificateur" a été défini, il est préférable d'utiliser la forme "identificateur" de "Valeur-Entier".

## 15 Notation du type Enuméré

15.1 Le type énuméré (voir le § 3.17) est dénoté par la notation "Type-Enuméré":

```
Type-Enuméré ::= ENUMERATED {Enumération}
Enumération ::=
    Nombre-Nommé |
    Nombre-Nommé, Enumération
```

*Remarque 1* – Chaque valeur a un identificateur, associé dans cette notation à un entier distinct. Il permet de contrôler la représentation de la valeur pour faciliter des extensions compatibles, mais les valeurs elles-mêmes ne sont pas censées avoir une sémantique d'entier.

*Remarque 2* – Les valeurs numériques des "Nombre-Nommé" de la partie droite de "Enumération" ne sont pas nécessairement ordonnées ni consécutives.

15.2 Pour chaque "Nombre-Nommé", "identificateur" et "Nombre-Signé" doivent être distincts de tous les autres "identificateur" et "Nombre-Signé" de "Enumération".

15.3 L'étiquette du type Enuméré est de classe "universal", numéro 10.

15.4 La valeur d'un type Enuméré est définie par la notation "Valeur-Enuméré":

```
Valeur-Enuméré ::= identificateur
```

## 16 Notation du type Réel

16.1 Le type Réel (voir le § 3.18) est dénoté par la notation "Type-Réel":

```
Type-Réel ::= REAL
```

16.2 Les valeurs du type Réel sont les valeurs PLUS-INFINITY et MINUS-INFINITY, ainsi que les nombres réels spécifiés par la formule suivante, qui comprend trois entiers, M, B et E:

$$M \times B^E$$

où M est appelée la mantisse, B la base et E l'exposant. M et E peuvent prendre n'importe quelle valeur entière, positive ou négative, alors que B ne peut prendre que les valeurs 2 ou 10. Toutes les combinaisons de M, B et E sont autorisées.

*Remarque 1* – Ce type peut véhiculer une représentation exacte de tout nombre pouvant être enregistré dans un matériel de virgule flottante standard, ainsi que tout nombre à représentation décimale finie par caractères.

*Remarque 2* – Le codage de ce type, qui est spécifié dans la Recommandation X.209, permet l'utilisation d'une base 2, 8 ou 16 avec une représentation binaire de valeurs réelles, et d'une base 10 avec une représentation par caractères. Le choix est laissé à l'émetteur.

16.3 L'étiquette du type Réel est de classe "universal", numéro 9.

16.4 La notation de définition d'une valeur d'un type Réel est "Valeur-Réel":

Valeur-Réel ::= Valeur-Réel-Numérique | Valeur-Réel-Spécial

Valeur-Réel-Numérique ::=

{Mantisse, Base, Exposant} | 0

Mantisse ::= Nombre-Signé

Base ::= 2 | 10

Exposant ::= Nombre-Signé

Valeur-Réel-Spécial ::= PLUS-INFINITY | MINUS-INFINITY

Pour la valeur "0" la seconde forme de "Valeur-Réel-Numérique" ne sera pas utilisée, mais seulement la forme "0".

## 17 Notation du type Chaîne binaire

17.1 Le type Chaîne binaire (voir le § 3.19) est dénoté par la notation "Type-Chaîne-Binaire":

Type-Chaîne-Binaire ::=

BIT STRING |  
BIT STRING {Liste-Bits-Nommés}

Liste-Bits-Nommés ::=

Bit-Nommé |  
Liste-Bits-Nommés, Bit-Nommé

Bit-Nommé ::=

identificateur (nombre) |  
identificateur (Valeur-Défini)

17.2 La "Liste-Bits-Nommés" n'est pas significative dans la définition d'un type. Elle est uniquement utilisée dans la notation de valeur spécifiée au § 17.8.

17.3 Le premier bit d'une Chaîne binaire a le numéro zéro. Le dernier bit d'une Chaîne binaire est appelé le bit de fin.

*Remarque* – Cette terminologie est utilisée pour spécifier la notation des valeurs et les règles de codage.

17.4 La "Valeur-Défini" est une dénotation d'une valeur non négative d'un type Entier, ou énuméré, ou par étiquetage ou par sous-type.

17.5 Chaque "nombre" ou "Valeur-Défini" figurant dans la "Liste-Bits-Nommés" doit avoir une valeur différente, qui est le numéro d'un bit distingué d'une valeur de chaîne binaire.

17.6 Chaque "identificateur" apparaissant dans la "Liste-Bits-Nommés" doit être différent.

*Remarque* – L'ordre des séquences de "Bits-Nommés" de la "Liste-Bits-Nommés" est sans importance.

17.7 L'étiquette de ce type est de classe "universal", numéro 3.

17.8 La valeur d'un type Chaîne binaire est définie par la notation "Valeur-Chaîne-Binaire":

Valeur-Chaîne-Binaire ::=

chaîne-b |  
chaîne-h |  
{Liste-Identificateurs} |  
{}

Liste-Identificateurs ::=

Identificateur |  
Liste-Identificateurs, identificateur

17.9 Chaque "identificateur" d'une "Valeur-Chaîne-Binaire" doit être le même qu'un "identificateur" de la séquence "Type-Chaîne-Binaire" à laquelle la valeur est associée.

17.10 L'utilisateur de la notation détermine, et peut indiquer par un commentaire, si la présence ou l'absence de bits zéro de fin est significative ou non.

*Remarque* – Les règles de codage permettent de transférer une chaîne de bits de n'importe quelles configuration et longueur.

17.11 Les notations "{Liste-Identificateurs}" et "{}" de "Valeur-Chaîne-Binaire" ne doivent pas être utilisées si la présence ou l'absence de bits zéro de fin est significative. Cette notation exprime une valeur de chaîne binaire avec des uns aux positions binaires spécifiées par les nombres correspondants aux séquences "identificateur" et des zéros à toutes les autres.

*Remarque* – La séquence "{}" sert à désigner une valeur de chaîne binaire qui ne contient aucun bit à un.

17.12 Lors de la spécification des règles de codage d'une chaîne binaire, les bits doivent être désignés par les termes **premier bit** et **bit de fin**, comme défini ci-dessus.

17.13 Lors d'une utilisation de la notation "chaîne-b", le **premier bit** est à gauche et le **bit de fin** est à droite.

17.14 Lors de l'utilisation de la notation "chaîne-h", le bit **de plus fort poids** de chaque chiffre hexadécimal correspond au premier bit (à gauche) de la chaîne binaire.

*Remarque* – Cette notation n'impose aucune contrainte quant à la manière dont les règles de codage convertissent une chaîne binaire en octets pour le transfert.

17.15 La notation "chaîne-h" ne peut être utilisé que si:

- a) la valeur de la chaîne binaire consiste en un multiple de quatre bits; ou
- b) la présence ou l'absence de bits zéro de fin n'est pas significative.

*Exemple:*

'A98A'H

et

'1010100110001010'B

sont deux notations possibles de la même valeur de chaîne binaire.

## 18 Notation du type Chaîne d'octets

18.1 Le type Chaîne d'octets (voir le § 3.20) est dénoté par la notation "Type-Chaîne-Octets":

Type-Chaîne-Octets ::= OCTET STRING

18.2 L'étiquette de ce type est de classe "universal", numéro 4.

18.3 La valeur d'un type Chaîne d'octets est défini par la notation "Valeur-Chaîne-Octets":

```
Valeur-Chaîne-Octets ::=  
    chaîne-binaire |  
    chaîne-h
```

18.4 Lors de la spécification des règles de codage d'une chaîne d'octets, les octets extrêmes sont désignés par les termes **premier octet et octet de fin**, et les bits extrêmes d'un octet sont désignés par les expressions **bit de plus fort poids et bit de plus faible poids**.

18.5 Lors de l'utilisation de la notation "chaîne-b", le bit le plus à gauche doit être le bit de plus fort poids du premier octet. Si la "chaîne-b" n'est pas un multiple de huit bits, elle doit être interprétée comme si elle contenait des bits zéro de fin additionnels, pour la compléter au prochain multiple de huit.

18.6 Lors de l'utilisation de la notation "chaîne-h", le chiffre hexadécimal le plus à gauche doit être le demi-octet le plus significatif du premier octet. Si la "chaîne-h" n'est pas un nombre pair de chiffres hexadécimaux, elle doit être interprétée comme si elle contenait un chiffre hexadécimal zéro de fin supplémentaire.

## 19 Notation du type Vide

19.1 Le type Vide (voir le § 3.21) est dénoté par la notation "Type-Vide":

```
Type-Vide ::= NULL
```

19.2 L'étiquette de ce type est de classe "universal", numéro 5.

19.3 La valeur du type vide est dénotée par la notation "Valeur-Vide":

```
Valeur-Vide ::= NULL
```

## 20 Notation des types Séquence

20.1 La notation pour définir un type Séquence (voir le § 3.22) à partir d'autres types est "Type-Séquence":

```
Type-Séquence ::=  
    SEQUENCE {Liste-Types-Elément} |  
    SEQUENCE { }
```

```
Liste-Type-Eléments ::=  
    Type-Elément |  
    Liste-Types-Eléments, Type-Elément
```

```
Type-Elément ::=  
    Type-Nommé |  
    Type-Nommé OPTIONAL |  
    Type-Nommé DEFAULT Valeur |  
    COMPONENTS OF Type
```

20.2 Le "Type" de la quatrième séquence possible dérivée de "Type-Elément" doit être un type Séquence. La notation "COMPONENTS OF Type" doit être utilisée pour définir l'inclusion, à ce point de la "Liste-Types-Eléments", de toutes les séquences "Type-Elément" apparaissant dans le type dénoté.

*Remarque* – Cette transformation doit être logiquement achevée avant de vérifier si les exigences des paragraphes suivants sont satisfaites.

20.3 Pour chaque série d'un ou plusieurs "Types-Eléments" consécutifs marqués comme OPTIONAL ou DEFAULT, les étiquettes de ces "Types-Eléments" et de tout "Type-Elément" immédiatement consécutif doivent être distinctes (voir le § 26).

20.4 Si "OPTIONAL" ou "DEFAULT" figurent, les valeurs correspondantes peuvent être omises dans la valeur du nouveau type ainsi que dans les informations transférées après codage.

*Remarque 1* – La notation de la valeur peut être ambiguë dans ce cas, sauf si des séquences "identificateur" figurent dans chaque Type-Nommé.

*Remarque 2* – Les règles de codage garantissent que le codage de la valeur d'une séquence dans laquelle une valeur d'élément "DEFAULT" ou "OPTIONAL" est omise, est le même que celui de la valeur d'une séquence dont l'élément correspondant a été omis dans la définition. Cette caractéristique peut être utile pour la définition de sous-ensembles.

20.5 Si "DEFAULT" figure, l'omission d'une valeur de ce type sera strictement équivalente à l'insertion de la valeur définie par "Valeur", qui doit être une spécification de valeur qui est valide pour le type défini par "Type" dans la séquence "Type-Nommé".

20.6 Tous les "identificateurs" (s'ils existent) de toutes les séquences "Type-Nommé" de la "Liste-Types-Eléments" doivent être distincts.

20.7 Tous les types Séquence ont une étiquette, qui est le numéro 16 de la classe "universal":

*Remarque* – Les types Séquence-de ont la même étiquette (voir le § 21.3).

20.8 La notation de définition de la valeur d'un type Séquence est "Valeur-Séquence":

Valeur-Séquence ::=

{Liste-Valeurs-Eléments} |  
{ }

Liste-Valeurs-Eléments ::=

Valeur-Nommé |  
Liste-Valeurs-Eléments, Valeur-Nommé

20.9 La notation "{}" sera utilisée uniquement si:

- toutes les séquences "Type-Elément" de "Type-Séquence" sont marquées DEFAULT ou OPTIONAL, et que toutes les valeurs sont omises; ou
- la notation du type était "SEQUENCE {}".

20.10 Seule une "Valeur-Nommé" doit figurer pour chaque "Type-Nommé" dans le "Type-Séquence" qui n'est pas marqué OPTIONAL ou DEFAULT, et les valeurs doivent être dans le même ordre que les séquences "Type-Nommé" correspondantes.

*Remarque* – L'emploi de séquences "Type-Nommé" qui ne contiennent pas d'identificateur n'est pas interdit, mais il risque de rendre ambiguë la notation de la valeur si "OPTIONAL" ou "DEFAULT" est utilisé.

## 21 Notation des types Séquences-de

21.1 La notation pour définir un type Séquence-de (voir le § 3.23) à partir d'un autre type est "Type-Séquence-De":

Type-Séquence-De ::=

SEQUENCE OF Type |  
SEQUENCE

21.2 La notation "SEQUENCE" est synonyme de la notation "SEQUENCE OF ANY" (voir le § 27).

21.3 L'étiquette de tous les types Séquence-de est de classe "universal", numéro 16.

*Remarque* – Les types de Séquence ont la même étiquette (voir le § 20.7).

21.4 La notation de définition d'une valeur d'un type Séquence-de est "Valeur-Séquence-De":

Valeur-Séquence-De ::= {Liste-Valeurs} | { }

Liste-Valeurs ::=

Valeur |  
Liste-Valeurs, Valeur

La notation "{}" est utilisée quand il n'y a pas de valeurs composantes dans la valeur séquence-de.

21.5 Chaque séquence "Valeur" de la "Liste-Valeurs" doit être la notation d'une valeur du "Type" spécifié dans le "Type-Séquence-De".

*Remarque* - L'ordre de ces valeurs peut avoir une signification sémantique.

## 22 Notation des types Ensemble

22.1 La notation pour définir un type Ensemble (voir le § 3.24) à partir d'autres types est le "Type-Ensemble":

Type-Ensemble ::=

$$\text{SET \{Liste-Types-Eléments\} | } \\ \text{SET \{ \}}$$

"Liste-Types-Eléments" est spécifié au § 20.1

22.2 Le quatrième "Type" possible du "Type-Elément" (voir le § 20.1) doit être un type ensemble. La notation "COMPONENTS OF type" doit être utilisée pour définir l'inclusion de toutes les séquences "Type-Elément" figurant dans le type dénoté.

*Remarque* – Cette transformation doit être logiquement achevée avant de vérifier si les exigences des paragraphes suivants sont satisfaites.

22.3 Les types "Type-Elément" d'un type ensemble doivent tous avoir des étiquettes différentes (voir le § 26).

22.4 Les § 20.4, 20.5 et 20.6 s'appliquent aussi aux types ensemble.

22.5 Tous les types ensemble ont une étiquette, qui est le numéro 17 de la classe "universal".

*Remarque* – Les types Ensemble-de ont la même étiquette (voir le § 23.3).

22.6 Dans un type ensemble, l'ordre des valeurs n'a pas de signification sémantique.

22.7 La notation de définition de la valeur d'un type ensemble est "Valeur-Ensemble":

Valeur-Ensemble ::= {Liste-Valeurs-Eléments} | {}

"Liste-Valeurs-Eléments" est spécifié au § 20.8.

22.8 La "Valeur-Ensemble" ne doit être "{}" que si:

- toutes les séquences "Type-Elément" du "Type-Ensemble" sont marquées "DEFAULT" ou "OPTIONAL" et si toutes les valeurs sont omises; ou
- la notation du type est "SET {}".

22.9 Il doit y avoir une "Valeur-Nommé" pour chaque "Type-Nommé" du "Type-Ensemble" qui n'est pas marqué "OPTIONAL" ou "DEFAULT".

*Remarque 1* – Ces "Valeurs-Nommés" peuvent figurer dans n'importe quel ordre.

*Remarque 2* – L'utilisation de séquences "Type-Nommé" ne comportant pas d'identificateur, n'est pas interdite mais elle risque de rendre ambiguë la notation de la valeur.

## 23 Notation des types Ensemble-de

23.1 La notation pour définir un type Ensemble-de (voir le § 3.25) à partir d'un autre type est "Type-Ensemble-De":

Type-Ensemble-De ::= SET OF Type |  
SET

23.2 La notation "SET" est synonyme de la notation "SET OF ANY" (voir le § 27).

23.3 Tous les types Ensemble-de ont une étiquette de classe "universal", numéro 17.

*Remarque* – Les types Ensemble ont la même étiquette (voir le § 22.5).

23.4 La notation de définition d'une valeur d'un type Ensemble-de est la "Valeur-Ensemble-De":

Valeur-Ensemble-De ::= {Liste-Valeurs} | {}

"Liste-Valeurs" est spécifiée au § 21.4.

La notation "{}" est utilisée quand il n'y a aucune valeur composante dans les valeurs d'Ensemble-de.

23.5 Chaque séquence "Valeur" de la "Liste-Valeurs" doit être la notation d'une valeur du "Type" spécifiée dans le "Type-Ensemble-De".

*Remarque 1* – L'ordre de ces valeurs n'a aucune signification sémantique.

*Remarque 2* – Aucune règle de codage n'est requise pour préserver l'ordre de ces valeurs.

## 24 Notation des types Choix

24.1 La notation pour définir un type Choix (voir le § 3.27) à partir d'autres types est le "Type-Choix":

Type-Choix ::= CHOICE {Liste-Types-Possibles}

Liste-Types-Possibles ::=

Type-Nommé |  
Liste-Types-Possibles, Type-Nommé

*Remarque 1* – Le type possible choisi est codé de façon qu'il ne puisse pas être distingué d'un "Type" composé seulement du "Type" contenu dans ce type possible.

*Remarque 2* – Le fait de spécifier un "Type-Choix" par un seul "Type-Nommé" dans la "Liste-Types-Possibles" équivaut à utiliser directement le "Type" dans le "Type-Nommé" et ne peut pas être distingué dans le codage de la valeur.

24.2 Les types définis dans la "Liste-Types-Possibles" doivent avoir des étiquettes distinctes (voir le § 26).

24.3 L'étiquette du type Choix doit être considérée comme variable. Lorsqu'une valeur est choisie, l'étiquette devient égale à l'étiquette de "Type" du "Type-Nommé" de la "Liste-Types-Possibles" dans laquelle la valeur est prise.

24.4 Lorsque ce type est utilisé dans un cas où la présente Recommandation exige d'employer des types comportant des étiquettes distinctes (voir les § 20.3, 22.3 et 24.2, les étiquettes de tous les types définis dans la "Liste-Types-Possibles" doivent être différentes de celles des autres types (voir le § 24). Les exemples suivants illustrent cette exigence. Les exemples 1 et 2 sont des utilisations correctes de la notation. L'exemple 3 n'est pas correct parce que les étiquettes des types d et f, et e et g sont identiques.

*Exemple 1:*

```
A ::= CHOICE
    { b B,
      c NULL }

B ::= CHOICE
    { d [0] NULL,
      e [1] NULL }
```

Exemple 2:

```
A ::= CHOICE
    { b B,
      c C }

B ::= CHOICE
    { d [0] NULL,
      e [1] NULL }

C ::= CHOICE
    { f [2] NULL,
      g [3] NULL }
```

Exemple 3:

```
(INCORRECT)
A ::= CHOICE
    { b B,
      c C }

B ::= CHOICE
    { d [0] NULL,
      e [1] NULL }

C ::= CHOICE
    { f [0] NULL,
      g [1] NULL }
```

24.5 Tous les "identificateurs" (le cas échéant) figurant dans toutes les séquences "Type-Nommé" de "Liste-Types-Possibles" doivent être distincts.

24.6 Lorsque ce type est utilisé dans un cas où la présente Recommandation exige l'emploi de "Type-Nommé" avec des "identificateurs" distincts, les "identificateurs" (le cas échéant) de tous les "Types-Nommés" de la "Liste-Types-Possibles" doivent être différents de ceux (le cas échéant) des autres "Types-Nommés".

24.7 La notation pour exprimer la valeur d'un type Choix est "Valeur-Choix":

Valeur-Choix ::= Valeur-Nommé

24.8 Si la "Valeur-Nommé" contient un "identificateur", ce doit être la notation d'une valeur de ce type de la "Liste-Types-Possibles", qui est nommée par le même "identificateur". Si la "Valeur-Nommé" ne contient pas d'"identificateur", ce doit être la notation d'une valeur de l'un des types de la "Liste-Types-Possibles" qui ne sont pas nommés par un "identificateur".

*Remarque* – L'omission d'un "identificateur" dans le "Type-Nommé" risque de rendre la notation de la valeur ambiguë.

## 25 Notation des types Sélection

25.1 Un "Type-Nommé" apparaissant dans la "Liste-Types-Possibles" d'un "Type-Choix" peut être dénoté par la notation "Type-Sélection":

Type-Sélection ::= Identificateur < Type

où "Type" fait référence au "Type-Choix" et "identificateur" à "identificateur" du "Type-Nommé".

*Remarque* – "Type-Sélection" peut être utilisée soit comme un "Type-Nommé" et dans ce cas "identificateur" est utilisé dans la notation de la valeur, soit comme "Type" dans un "Type Nommé" et dans ce cas, son "identificateur" n'est pas employé.

25.2 La notation d'une valeur d'un Type sélection est "Valeur-Sélection":

Valeur-Sélection ::= Valeur-Nommé

où la "Valeur-Nommé" contient l'identificateur qui apparaît dans le "Type-Sélection" correspondant si et seulement si ce "Type-Sélection" est utilisé comme "Type-Nommé".

## 26 Notation des types Etiquetés

Un type Etiqueté (voir le § 3.26) est un nouveau type, qui est isomorphe à un type existant, mais avec une étiquette différente. Dans tous les systèmes de codage, une valeur du nouveau type peut être distinguée d'une valeur de l'ancien type. Le type Etiqueté est principalement utilisé lorsque la présente Recommandation exige l'emploi de types à étiquettes distinctes. (Voir les § 20.3, 22.3, 24.2, 24.4 et 27.6).

*Remarque* – Lorsqu'un protocole détermine que les valeurs de plusieurs types de données peuvent être transmises à tout moment, des étiquettes distinctes peuvent être nécessaires pour permettre un décodage correct de chaque valeur par le destinataire.

26.1 La notation d'un type Etiqueté est "Type-Etiqueté":

Type-Etiqueté ::=

Type-Etiquette |  
Type-Etiquette IMPLICIT |  
Type-Etiquette EXPLICIT

Etiquette ::= [Classe Numéro-Classe]

Numéro-Classe ::=

numéro |  
Valeur-Défini |

Classe ::=

UNIVERSAL |  
APPLICATION |  
PRIVATE |  
vide

26.2 La "Valeur-Défini" doit être une dénotation d'une valeur non négative de type Entier, ou d'un type dérivé d'un type Entier par étiquetage.

26.3 Le nouveau type est isomorphe à l'ancien, mais a une étiquette de classe "Classe" et de numéro "Numéro-Classe", sauf si la "Classe" est "vide", quand l'étiquette est de classe spécifique au contexte, auquel cas elle a seulement un numéro "Numéro-Classe".

26.4 La "Classe" ne doit pas être "UNIVERSAL" sauf pour les types définis dans la présente Recommandation.

*Remarque* – L'utilisation des étiquettes de classe "universal" fait l'objet d'agréments périodiques de la part de l'ISO et du CCITT.

26.5 Si la "Classe" est "APPLICATION" la même "Etiquette" ne doit pas être utilisée à nouveau dans le même module.

26.6 Si la "Classe" est "PRIVATE", l'"Etiquette" est disponible pour une utilisation spécifique à une entreprise.

26.7 La construction de l'étiquetage spécifie un étiquetage explicite si l'une des conditions suivantes est remplie:

- a) la possibilité "Type-Etiquette EXPLICIT" est utilisée;
- b) la possibilité "Type-Etiquette" est utilisée et la valeur de l'"Etiquette-Défaut" du module est "EXPLICIT TAGS";
- c) la possibilité "Type-Etiquette" est utilisée, et la valeur de l'"Etiquette-Défaut" du module est "IMPLICIT TAGS", mais le type défini par "Type" est un type Choix ou un type Quelconque.

Dans tous les autres cas, la construction de l'étiquetage spécifie un étiquetage implicite.

26.8 Si la "Classe" est "vide", il n'est imposé à l'utilisation de l'"Etiquette" aucune autre limitation que celle impliquée par l'obligation d'étiquettes distinctes formulée dans les § 20.3, 22.3 et 24.2.

26.9 L'étiquetage implicite indique pour les règles de codage qui en offrent l'option, que l'identification explicite de l'étiquette du Type dans le "Type-Etiqueté" n'est pas nécessaire pour le transfert.

*Remarque* – Il peut être utile de conserver l'ancienne étiquette lorsqu'elle était de classe "universal", et qu'elle identifie donc de façon non ambiguë le type sans connaître la définition ASN.1 du nouveau type. Toutefois, l'utilisation d'IMPLICIT permet de réduire au minimum le nombre d'octets nécessaires lors d'un transfert. Un exemple de codage utilisant IMPLICIT est donné dans la Recommandation X.209.

26.10 La possibilité "IMPLICIT" ne doit pas être utilisée si le type défini par "Type" est un type Choix ou un type Quelconque.

26.11 La notation d'une valeur d'un "Type-Etiqueté" est "Valeur-Etiqueté":

Valeur-Etiqueté ::= Valeur

où "Valeur" est la notation d'une valeur du "Type" de "Type-Etiqueté".

*Remarque* – L'"Etiquette" n'apparaît pas dans cette notation.

## 27 Notation du type Quelconque

27.1 La notation d'un type Quelconque (voir le § 3.29) est "Type-Quelconque":

Type-Quelconque ::=

ANY |  
ANY DEFINED BY identificateur

*Remarque* – L'utilisation d'"ANY" dans une norme ISO ou une Recommandation du CCITT produit une spécification incomplète, si elle n'est pas complétée par une spécification additionnelle. La structure "ANY DEFINED BY" offre la possibilité de spécifier, dans un cas de communication, le type destiné au champ "ANY" et un pointeur qui désigne sa sémantique. Si les règles d'application suivantes sont respectées, "ANY" peut fournir une spécification complète. L'utilisation d'ANY sans la structure DEFINED BY est déconseillée.

27.2 La possibilité "DEFINED BY" doit seulement être utilisée quand le type quelconque, ou un type qui en dérive par étiquetage, est l'un des types composants d'un type séquence ou d'un type ensemble (le type contenant).

27.3 L'"identificateur" de la possibilité "DEFINED BY" doit également apparaître dans un "Type-Nommé" qui spécifie un autre composant, non optionnel, du type contenant. Le "Type-Nommé" doit être un type Entier ou un type Enuméré ou un type Identificateur d'objet ou un type qui en dérive par étiquetage ou sous-typage.

27.4 Quand le "Type-Nommé" est un type Entier ou Enuméré, ou un type qui en dérive par étiquetage ou sous-typage, le document employant la notation "DEFINED BY" doit contenir, ou citer de façon explicite, une liste unique qui spécifie le type ASN.1 à véhiculer par ANY pour chaque valeur autorisée du type Entier. Une telle liste, et une seule, doit figurer dans chaque communication du type contenant.

27.5 Quand le "Type-Nommé" est un type Identificateur d'objet ou un type qui en dérive par étiquetage, il faut que des registres associent, à chaque valeur d'Identificateur d'objet associée, un type ASN.1 unique (qui peut être un type CHOICE) qui doit être véhiculé par le champ d'ANY.

*Remarque 1* – Le nombre de registres associant une valeur d'Identificateur d'objet à un type ASN.1 à cet effet, peut être quelconque.

*Remarque 2* – L'enregistrement des valeurs, effectué à des fins d'interconnexion de systèmes ouverts, devrait se faire dans les normes ISO et les Recommandations du CCITT utilisant la notation. Quand l'enregistrement par une Autorité d'enregistrement internationale différente est prévue pour une occurrence d'ANY DEFINED BY, ceci doit être indiqué dans le document utilisant la notation.

*Remarque 3* – La principale différence entre la définition d'un Entier et celle d'un Identificateur d'objet, est que l'utilisation d'un Entier se réfère à une liste unique, contenue dans la norme ou recommandation qui l'utilise, alors que l'utilisation de l'Identificateur d'objet permet un ensemble non clos de types déterminés par une autorité habilitée à affecter des identificateurs d'objets.

27.6 Ce type a une étiquette indéterminée et ne doit pas être utilisé lorsque la présente Recommandation exige des étiquettes distinctes (voir les § 20.3, 22.3, 24.2 et 24.4).

27.7 La notation de la valeur d'un type Quelconque doit être définie en utilisant ASN.1 et est "Valeur-Quelconque":  
Valeur-Quelconque ::= Valeur-Type  
où "Type" est la notation du type choisi et "Valeur" la notation d'une valeur de ce type.

## 28 Notation du type Identificateur d'objet

28.1 Le type Identificateur d'objet (voir le § 3.34) est dénoté par la notation "Type-Identificateur-Objet".

Type-Identificateur-Objet ::=

OBJECT IDENTIFIER

28.2 L'étiquette de ce type est de classe "universal", numéro 6.

28.3 La notation de valeur d'un Identificateur d'objet est "Valeur-Identificateur-Objet":

Valeur-Identificateur-Objet ::=

{ Liste-Composants-Id-Ob } |  
{ Valeur-Défini Liste-Composants-Id-Ob }

Liste-Composants-Id-Ob ::=

Composant-Id-Ob |  
Composant-Id-Ob Liste-Composants-Id-Ob

Composant-Id-Ob ::=

Forme-Nom |  
Forme-Numéro |  
Forme-Nom-Et-Numéro

Forme-Nom ::= identificateur

Forme-Numéro ::= numéro | Valeur-Défini

Forme-Nom-Et-Numéro ::=

identificateur (Forme-Numéro)

28.4 La "Valeur-Défini" de "Forme-Numéro" doit dénoter une valeur du type Entier ou Enuméré, ou d'un type qui en dérive par étiquetage ou sous-typage.

28.5 La "Valeur-Défini" de la "Valeur-Identificateur-Objet" doit dénoter une valeur du type Identificateur d'objet ou d'un type qui en dérive par étiquetage.

28.6 La "Forme-Nom" doit être utilisée uniquement pour les composants de l'Identificateur d'objet dont la valeur numérique et l'identificateur sont spécifiés dans les annexes B à D, et doit être l'un des identificateurs spécifiés dans ces annexes.

28.7 Le "Numéro" de la "Forme-Numéro" doit être la valeur numérique affectée au composant de l'Identificateur d'objet.

28.8 L'"identificateur" de la "Forme-Nom-Et-Numéro" doit être spécifié quand une valeur numérique est affectée au composant de l'Identificateur d'objet.

*Remarque* – Les autorités affectant des valeurs numériques au composant de l'Identificateur d'objet sont indiquées dans les annexes de la présente Recommandation.

28.9 La sémantique d'une valeur d'Identificateur d'objet est définie par référence à un **arbre d'identificateurs d'objet**. Un arbre d'identificateurs est un arbre dont la racine correspond à la présente Recommandation, et les noeuds aux autorités administratives responsables de l'affectation des arcs issus de chaque noeud. Chaque arc est étiqueté par un composant de l'Identificateur d'objet qui est une valeur numérique. Chaque objet informationnel à identifier est affecté à un noeud et un seul (normalement une feuille) et aucun autre objet informationnel, de type identique ou différent n'est affecté à ce noeud. Un objet informationnel est ainsi identifié d'une façon unique et non ambiguë par la

séquence des valeurs numériques (composants de l'Identificateur d'objet) libellant des arcs d'un chemin allant de la racine au noeud attribué à cet objet informationnel.

*Remarque* – Les valeurs d'Identificateur d'objet contiennent au moins deux composants d'Identificateur d'objet, comme spécifié dans les annexes B à D.

28.10 Sémantiquement, la valeur d'un Identificateur d'objet est une liste ordonnée de valeurs de composants d'identificateurs d'objet. Chaque valeur de composant d'Identificateur d'objet identifie un arc de l'arbre des identificateurs d'objets à partir de la racine. La dernière valeur de composant d'Identificateur d'objet identifie un arc allant à un noeud auquel un objet informationnel a été affecté. C'est cet objet informationnel qui est identifié par la valeur de l'Identificateur d'objet. La partie significative du composant d'Identificateur d'objet est la "Forme-Nom" ou "Forme-Numéro" à laquelle il se ramène, et qui donne la valeur numérique de ce composant d'identificateurs d'objet.

*Remarque* – En général, un objet informationnel est une classe d'informations (par exemple un format de fichier) plutôt qu'une occurrence d'une telle classe (par exemple un fichier particulier). C'est donc à la classe d'information (définie par une spécification à laquelle il peut être fait référence) plutôt qu'à l'élément d'information lui-même qu'est effectué un emplacement dans l'arbre.

28.11 Quand la "Valeur-Identificateur-Objet" inclut une "Valeur-Défini", la liste des composants d'identificateurs d'objet à laquelle elle se réfère est accolée en préfixe aux composants figurant explicitement dans la valeur.

*Exemples:* Avec les identificateurs affectés comme spécifié dans l'annexe B, les valeurs:

{iso norme 8571 pci (1)}

et

{1 0 8571 1}

identifieraient chacune un objet, "pci", défini dans la Norme ISO 8571.

Avec la définition additionnelle suivante:

ftam OBJECT IDENTIFIER ::=

{iso norme 8571 }

la valeur suivante est également équivalente aux valeurs ci-dessus:

{ftam pci (1)}

*Remarque* – Chaque fois qu'une Recommandation du CCITT, une Norme ISO ou un autre document affectent des valeurs du type OBJECT IDENTIFIER, à des objets informationnels, une annexe ou un appendice devrait résumer les affectations qui y sont faites. Il est également recommandé que la même autorité puisse affecter un objet informationnel des valeurs du type OBJECT IDENTIFIER et des valeurs du type "Descripteur-Objet".

## 29 Notation des types Chaîne de caractères

29.1 La notation de dérivation d'un type Chaîne de caractères (voir le § 3.12 et la section 2) est:

Type-chaîne-Caractères ::= dénotation-type

où "dénotation-type" est un des noms du type de Chaîne de caractères indiqués dans la section 2.

29.2 L'étiquette de chaque type Chaîne de caractères est spécifiée dans la section 2.

29.3 La notation d'une valeur de Chaîne de caractères est:

Valeur-Chaîne-Caractères ::= chaîne-c

La définition du type Chaîne de caractères détermine les caractères qui apparaissent dans la "chaîne-c".

## 30 Notation des types définis dans la section 3

30.1 La notation utilisée pour la dénotation d'un type défini dans la section 3 de la présente Recommandation est:

Type-Utile ::= dénotation-type

où "dénotation-type" est l'un de ceux définis dans la section 3 en notation ASN.1.

- 30.2 L'étiquette de chaque "Type-Utile" est spécifiée dans la section 3.
- 30.3 La notation d'une valeur d'un "Type-Utile" est spécifiée dans la section 3.

## SECTION 2 – TYPES CHAÎNE DE CARACTÈRES

### 31 Définition des types Chaîne de caractères

Ce chapitre définit des types dont les valeurs distinguées sont des séquences de zéro, un ou plusieurs caractères d'un certain jeu de caractères.

31.1 Le type est défini en spécifiant:

- a) l'étiquette affectée au type;
- b) un nom par lequel la définition du type peut être dénotée;
- c) les caractères du jeu de caractères utilisés pour définir le type, par référence à une table donnant les symboles de ces caractères, ou par référence à un numéro d'enregistrement dans le Registre international ISO des Jeux de caractères codés, à utiliser avec des séquences d'échappement.

Le nom b) ci-dessus peut être utilisé comme "dénotation-type" dans la notation ASN.1 (voir le § 29).

31.2 Le tableau 6/X.208 donne la liste des noms par lesquels chacune de ces définitions de type peut être désignée, le numéro de l'étiquette de classe "universal" affecté à ce type, le numéro d'enregistrement ou celui des tableaux ci-après qui le définit et, si nécessaire, l'identification d'une remarque concernant cette ligne du tableau. Quand un synonyme est défini dans la notation, il est indiqué entre parenthèses.

*Remarque* – L'étiquette affectée aux types Chaîne de caractères identifie le type de façon non ambiguë. A noter toutefois que si la notation ASN.1 est utilisée pour définir de nouveaux types à partir de ce type (en particulier en utilisant IMPLICIT), il risque de devenir impossible de reconnaître ces types sans avoir connaissance de la définition du type ASN.1.

31.3 Le tableau 4/X.208 donne les caractères pouvant figurer dans le type Chaîne-Numérique.

TABLEAU 4/X.208

#### Chaîne-Numérique

Nom	Symbole
Chiffres	0, 1, . . . 9
Espace	(espace)

**TABLEAU 5/X.208**

**Chaîne-Imprimable**

Nom	Symbole
Lettres majuscules	A, B, ... Z
Lettres minuscules	a, b, ... z
Chiffres	0, 1, ... 9
Espace	(espace)
Apostrophe	'
Parenthèse gauche	(
Parenthèse droite	)
Signe plus	+
Virgule	,
Trait d'union	-
Point	.
Barre oblique	/
Deux-points	:
Signe égal	=
Point d'interrogation	?

**TABLEAU 6/X.208**

**Liste des types de Chaîne de caractères**

Nom de dénotation du type	Numéro dans la classe universelle	Numéro d'enregistrement (voir ISO 2375) ou numéro du tableau précédent, donnant la définition	Remarques
Chaîne-Numérique	18	tableau 4	1
Chaîne-Imprimable	19	tableau 5	1
Chaîne-Télétext (Chaîne T61)	20	87, 102, 103, 106, 107 + SPACE + DELETE	2
Chaîne-Vidéotext	21	1, 72, 73, 102, 108, 128, 129 + SPACE + DELETE	3
Chaîne-Visible (Chaîne-ISO646)	26	2 + SPACE	
Chaîne-IA5	22	1, 2 + SPACE + DELETE	
Chaîne-Graphique	25	Tous les jeux G + SPACE	
Chaîne-Générale	27	Tous les jeux G et C + SPACE + DELETE	

*Remarque 1* – La typographie, la taille, la couleur, la graisse ainsi que les autres caractéristiques d'impression ou d'affichage ne sont pas significatives.

*Remarque 2* – Les entrées correspondants à ces numéros d'enregistrement renvoient à la Recommandation T.61, pour les règles concernant leur utilisation.

*Remarque 3* – Les entrées correspondants à ces numéros d'enregistrement donnent les fonctions définies dans les Recommandations T.100 et T.101.

31.4 Le tableau 5/X.208 donne la liste des caractères qui peuvent apparaître dans le type Chaîne-Imprimable.

31.5 La notation de ces types est "chaîne-c".

*Remarque* – Cette notation peut seulement être utilisée avec un dispositif capable d'afficher ou d'imprimer les caractères figurant dans la valeur. Dans les autres cas, la notation de la valeur n'est pas définie.

31.6 Dans tous les cas, la plage des caractères peut être restreinte par un commentaire, mais ne peut pas être étendue.

## SECTION 3 – DÉFINITIONS UTILES

Cette section contient des définitions qui devraient être utiles dans nombre d'applications.

*Remarque* – Il est prévu de compléter cette section pour prendre en compte d'autres types de données comme les diagnostics, les informations d'authentification, les informations comptables, les paramètres de sécurité, etc.

La notation de la valeur et la définition de la sémantique des types définis dans cette section sont dérivées d'une définition de type utilisant la notation ASN.1. Cette définition peut être citée par des normes ou recommandations définissant les règles de codage à appliquer pour spécifier le codage de ces types.

### 32 Heure généralisée

32.1 Ce type est dénoté par le nom:

Heure-Généralisée

32.2 Ce type est composé de valeurs représentant:

- a) une date, telle que définie dans la Norme ISO 2014 (Représentation numérique des dates);
- b) une heure, à une des précisions définies dans le § 2 de la Norme ISO 3307 (Représentation de l'heure);
- c) le facteur de décalage horaire local, tel qu'il est défini dans la Norme ISO 4031 (Représentation des différences d'heure).

32.3 Ce type peut être défini en ASN.1 comme suit:

Heure-Généralisée ::=

[UNIVERSAL 24] IMPLICIT  
Chaîne-Visible

les valeurs de "Chaîne-Visible" étant limitées à l'une des chaînes de caractères suivantes:

- a) une chaîne représentant la date, comme spécifié dans ISO 2014, l'année étant représentée par quatre chiffres, le mois par deux chiffres et le jour par deux chiffres, sans séparateur, suivie d'une chaîne représentant l'heure comme spécifié dans ISO 3307, sans séparateur autre que la virgule ou le point décimal, (comme prévu aux § 2.3, 2.4 et 2.5 de la Norme ISO 3307) et sans lettre z à la fin (comme prévu au § 3 de la Norme ISO 3307);
- b) les caractères de a) ci-dessus, suivis d'une lettre Z majuscule,
- c) les caractères de a) ci-dessus, suivis d'une chaîne représentant un décalage horaire local, comme spécifié dans ISO 4031, sans séparateur.

Dans le cas a), l'heure représente l'heure locale. Dans le cas b), l'heure représente l'heure universelle coordonnée (heure UTC). Dans le cas c), la partie de la chaîne formée comme dans le cas a) représente l'heure locale ( $t_1$ ), le décalage horaire ( $t_2$ ) permettant de déterminer l'heure UTC comme suit:

$$\text{Heure UTC} = t_1 - t_2$$

*Exemples:*

Cas a)                      19851106210627.3  
Heure locale 21 heures 6 minutes 27,3 secondes,  
le 6 novembre 1985.

- |        |   |
|--------|---|
| Cas b) | 19851106210627.3Z<br>Heure UTC correspondant à l'heure ci-dessus.   |
| Cas c) | 19851106210627.3 – 0500<br>Même heure locale que dans l'exemple a),<br>avec un décalage horaire de 5 heures<br>sur l'heure UTC. |

- 32.4 L'étiquette doit être conforme à la définition donnée au § 32.3
- 32.5 La notation de valeur doit être celle de la "Chaîne-Visible" définie au § 32.3

### 33 Heure universelle

- 33.1 Ce type est dénoté par le nom:

Heure-UTC

- 33.2 Le type est composé de valeurs représentant:

- a) une date;
- b) une heure, à la précision de la minute ou de la seconde;
- c) (facultativement) le décalage de l'heure locale par rapport à l'heure universelle coordonnée.

- 33.3 Le type peut être défini en ASN.1 comme suit:

Heure-UTC ::=

[UNIVERSAL 23] IMPLICIT  
Chaîne-Visible

les valeurs de la "Chaîne-Visible" étant limitées aux chaînes de caractères qui sont la juxtaposition de:

- a) six chiffres YYMMDD où YY sont les deux chiffres de plus faible poids de l'année chrétienne, MM est le mois (janvier étant 01) et DD le jour du mois (01 à 31); et
- b) soit:
  - 1) quatre chiffres hhmm où hh est l'heure (00 à 23) et mm les minutes (00 à 59); ou
  - 2) six chiffres hhhmss où hh et mm sont comme en 1) ci-dessus, et ss sont les secondes (00 à 59); et
- c) soit:
  - 1) le caractère Z; ou
  - 2) le caractère + ou le caractère –, suivi de hhmm, hh représentant les heures et mm les minutes.

Les deux possibilités de b) ci-dessus permettent des précisions variables dans la spécification de l'heure.

Dans la possibilité c) 1), l'heure est l'heure UTC, dans la possibilité c) 2), l'heure  $t_1$  spécifiée par a) et b) ci-dessus est l'heure locale; le décalage horaire ( $t_2$ ) spécifié par c) 2) ci-dessus permet de déterminer l'heure UTC comme suit:

l'heure UTC est  $t_1 - t_2$

*Exemple:* Si l'heure locale est 7 heures du matin, l'heure universelle coordonnée midi, le 2 janvier, la valeur est l'une des suivantes:

Heure-UTC "8201021200Z"  
Heure-UTC "8201020700-0500"

- 33.4 L'étiquette doit être définie comme au § 33.3.
- 33.5 La notation de la valeur doit être celle de la "Chaîne-Visible" définie au § 33.3.

### 34 Le type Externe

34.1 La notation d'un type Externe (voir le § 3.30) est "Type-Externe":

Type-Externe ::= EXTERNAL

34.2 Le type est composé des valeurs représentant:

- a) le codage d'une valeur donnée qui peut, mais pas nécessairement, être la valeur d'un seul type de données ASN.1;
- b) les informations d'identification déterminant la sémantique et les règles de codage;
- c) (facultativement) un descripteur d'objet décrivant l'objet.

Le descripteur d'objet facultatif ne doit pas figurer, sauf autorisation explicite par un commentaire accompagnant l'utilisation de la notation EXTERNAL.

34.3 Le type EXTERNAL permet d'inclure toute valeur de données provenant d'un ensemble identifié de valeurs de données.

*Remarque 1* – La spécification de cet ensemble de valeurs de données, leur sémantique, l'affectation d'un identificateur d'objet et (facultativement) d'un descripteur d'objet, ainsi que la diffusion de ces informations à tous les partenaires en communication est appelée **enregistrement d'une syntaxe abstraite**. Cet enregistrement peut être réalisé par toute autorité habilitée à affecter une valeur d'OBJECT IDENTIFIER comme spécifié dans les annexes B à D.

*Remarque 2* – Un ensemble de valeurs de données enregistré comme syntaxe abstraite (avec les règles de codage associées), n'est bien formé que si le codage de chaque valeur de données s'identifie de lui-même parmi l'ensemble des codages de valeurs de données. Quand une syntaxe abstraite est définie en ASN.1, cette auto-identification est assurée par l'étiquetage. Quand une syntaxe abstraite n'est pas bien formée, l'utilisation d'un canal de communication risque d'être dépendante du contexte ou de conduire à une ambiguïté.

34.4 Le type EXTERNAL peut être défini en ASN.1 comme suit:

**EXTERNAL ::= [UNIVERSAL 8] IMPLICIT SEQUENCE**

<b>{ dénotation-directe</b>	<b>OBJECT IDENTIFIER OPTIONAL,</b>
<b>dénotation-indirecte</b>	<b>INTEGER OPTIONAL,</b>
<b>descripteur-valeur-données</b>	<b>Descripteur-Objet OPTIONAL,</b>
<b>codage</b>	<b>CHOICE</b>
<b>{ type-simple-ASN1</b>	<b>[0] ANY,</b>
<b>aligné-octets</b>	<b>[1] IMPLICIT OCTET STRING,</b>
<b>arbitraire</b>	<b>[2] IMPLICIT BIT STRING }</b>

34.5 Quand la négociation de la Couche Présentation n'est pas pratiquée (avant l'accord sur la syntaxe de transfert) pour la valeur de cet "EXTERNAL", la "dénotation-directe OBJECT IDENTIFIER" doit figurer. Dans ce cas, l'identificateur de l'ensemble des valeurs de données est un identificateur d'objet qui dénote directement une syntaxe abstraite et occupe le champ "dénotation-directe OBJECT IDENTIFIER" de l'"EXTERNAL". En outre, l'enregistrement de la syntaxe abstraite définit également dans ce cas, les règles de codages (syntaxe de transfert) des valeurs de données, et la "dénotation-indirecte INTEGER" ne doit pas être incluse.

34.6 Si la négociation de la Couche Présentation est pratiquée pour la valeur de cet EXTERNAL, la "dénotation-indirecte INTEGER" doit figurer. Dans ce cas, l'identificateur de l'ensemble des valeurs de données est un entier qui identifie une utilisation d'une syntaxe abstraite. Cet entier est appelé **identificateur de contexte de présentation**, et occupe le champ "dénotation-indirecte INTEGER" de cet EXTERNAL. Si la négociation de la Couche Présentation a abouti, l'identificateur du contexte de présentation identifie également les règles de codage (syntaxe de transfert) des valeurs de données et la "dénotation-directe OBJECT IDENTIFIER" ne doit pas être incluse. Si la négociation de la Couche Présentation n'a pas abouti, il faut en plus une valeur d'identificateur d'objet pour identifier les règles de codage (syntaxe de transfert) utilisées pour le codage. L'utilisation éventuelle de la négociation de la Couche Présentation, et le transport (autorisé ou impératif) de cette valeur par l'élément "dénotation-directe OBJECT IDENTIFIER" doivent être spécifiés dans un commentaire associé à l'utilisation de la notation "EXTERNAL"; sinon le champ ne doit pas figurer.

*Remarque 1* – Les § 34.5 et 34.6 ont pour effet de rendre obligatoire la présence d'une "dénotation-directe" ou "dénotation-indirecte".

*Remarque 2* – Quand la négociation de la Couche Présentation est pratiquée, mais n'a pas abouti, les deux dénnotations figurent.

34.7 Si la valeur de données est celle d'un seul type de données ASN.1, et si ses règles de codage sont les mêmes que celles du type de données "EXTERNAL" complet, l'émetteur doit alors utiliser l'une des options de "Codage" suivantes:

type-ASN1-unique  
aligné-octets  
arbitraire

comme option de la mise en oeuvre.

34.8 Si, après acceptation ou négociation du codage, le codage de la valeur de données est un nombre entier d'octets, l'émetteur doit choisir une des options de "Codage" suivantes:

aligné-octets  
arbitraire

comme option de la mise en oeuvre.

*Remarque* – Une valeur de données qui est une série de types ASN.1, et pour laquelle la syntaxe de transfert spécifie la concaténation simple de la chaîne d'octets produite en appliquant les règles de codage de base ASN.1 à chaque type ASN.1, relève de cette catégorie, et non de celle du § 34.7.

34.9 Si, après acceptation ou négociation du codage, le codage de la valeur de données n'est pas un nombre entier d'octets, le choix du "Codage" doit être:

arbitraire

34.10 Si l'option de "Codage" choisie est "type-ASN1-unique", le type ASN.1 doit remplacer le type ANY par une valeur égale de données à coder.

*Remarque* - La plage des valeurs pouvant se présenter dans ANY est déterminée par l'enregistrement de la valeur d'OBJECT IDENTIFIER associée à la "dénotation-directe" et/ou par la valeur d'INTEGER associée à la "dénotation-indirecte".

34.11 Si l'option de "Codage" choisie est "aligné-octets", la valeur de données doit être codée suivant la syntaxe de transfert acceptée ou négociée, et les octets qui en résultent forment la valeur de la chaîne d'octets.

34.12 Si l'option de "Codage" choisie est "arbitraire", la valeur de données doit être codée selon la syntaxe de transfert acceptée ou négociée, et le résultat forme la valeur de la chaîne binaire.

34.13 L'étiquette est telle que définie au § 34.4.

34.14 La notation de valeur est celle du type défini au § 34.4.

## **35 Le type Descripteur d'objet**

35.1 Ce type est dénoté par le nom:

Descripteur-Objet

35.2 Ce type consiste en un texte lisible par l'homme et servant à décrire un objet informationnel. Ce texte n'est pas une identification non ambiguë de l'objet informationnel, mais il ne devrait pas être courant qu'un texte identique décrive des objets informationnels différents.

*Remarque* - Il est recommandé que la valeur du type "OBJECT IDENTIFIER" et celle du type "Descripteur-Objet" soient affectées par la même autorité à un objet informationnel.

35.3 Ce type peut être défini en notation ASN.1 comme suit:

Descripteur-Objet ::=

[UNIVERSAL 7] IMPLICIT  
Chaîne-Graphique

La "Chaîne-Graphique" contient le texte décrivant l'objet informationnel.

35.4 L'étiquette est telle que définie au § 35.3.

35.5 La notation de valeur doit être celle de la "Chaîne-Graphique" définie au § 37.3.

## SECTION 4 – SOUS-TYPES

### 36 Notation des sous-types

36.1 Un sous-type est défini par la notation d'un type parent suivi d'une spécification appropriée de sous-type. La notation de spécification d'un sous-type est articulée en ensembles de valeurs de sous-type. Les valeurs du sous-type sont déterminées comme spécifié au § 36.7, en prenant la réunion logique de tous les ensembles de valeurs du sous-type.

36.2 La notation du sous-type ne doit pas être utilisée en sorte de produire un sous-type ne comportant aucune valeur.

36.3 La notation d'un sous-type est "Sous-type":

Sous-type ::=

Type-Parent Spécif-Sous-type |  
SET Limitation-Taille OF Type |  
SEQUENCE Limitation-Taille OF Type

Type-Parent ::= Type

36.4 Quand la notation "Spécif-Sous-type" suit la notation "Type-Sélection", le type parent est le "Type-Sélection" et non le "Type" de la notation "Type-Sélection".

36.5 Quand la notation "Spécif-Sous-Type" suit la notation d'un type "Ensemble-De" ou "Séquence-De", elle s'applique au "Type" de cette notation ensemble-de ou séquence-de et non au type ensemble-de ou séquence-de.

*Remarque* - La notation spéciale "SET Limitation-Taille OF" et "SEQUENCE Limitation-Taille OF" permet de disposer d'un autre mécanisme pour les cas simples (plus lisible que la notation du cas général). Les cas plus complexes nécessitent le mécanisme général.

36.6 La notation de spécification du sous-type est "Spécif-Sous-type":

Spécif-Sous-type ::=

(Ensemble-Valeurs-Sous-type Liste-Ensemble-Valeurs-Sous-type)

Liste-Ensemble-Valeurs-Sous-type ::=

“ | ”

Ensemble-Valeurs-Sous-type

Liste-Ensemble-Valeurs-Sous-type |

vide

36.7 Chaque "Ensemble-Valeurs-Sous-type" spécifie un certain nombre (éventuellement zéro) de valeurs du type parent, qui sont incluses dans le sous-type. Une valeur du type parent est une valeur du sous-type si et seulement si elle est incluse dans un ou plusieurs des ensembles de valeurs du sous-type. Le sous-type est ainsi formé à partir de la réunion logique des valeurs appartenant aux ensembles de valeurs de sous-type.

36.8 Différentes formes de notation sont données pour "Ensemble-Valeurs-Sous-type". Elles sont indiquées ci-après et leur syntaxe et sémantique sont définies au § 37. Comme spécifié au § 37 et résumé dans le tableau 7/X.208, certaines notations ne peuvent s'appliquer qu'à des types de parents particuliers.

TABLEAU 7/X.208

**Applicabilité des ensembles de valeurs de sous-types**

Type (ou dérivé d'un type par étiquetage)	Valeur unique	Sous-type contenu	Plage valeur	Limitation taille	Limitation alphabet	Sous-typage interne
Booléen	/	/	x	x	x	x
Entier	/	/	/	x	x	x
Énuméré	/	/	x	x	x	x
Réel	/	/	/	x	x	x
Identificateur d'objet	/	/	x	x	x	x
Chaîne binaire	/	/	x	/	x	x
Chaîne d'octets	/	/	x	/	x	x
Types chaîne de caractères	/	/	x	/	/	x
Séquence	/	/	x	x	x	/
Séquence-de	/	/	x	/	x	/
Ensemble	/	/	x	x	x	/
Ensemble-de	/	/	x	/	x	/
Quelconque	/	/	x	x	x	x
Choix	/	/	x	x	x	/

Ensemble-Valeurs-Sous-type ::=

Valeur-Unique |  
 Sous-Type-Contenu |  
 Plage-Valeur |  
 Alphabet-Autorisé | Limitation-Taille | Contraintes-Types-Interne

### 37 Ensembles de valeurs d'un sous-type

#### 37.1 Valeur unique

37.1.1 La notation de "Valeur-Unique" est:

Valeur-Unique ::= Valeur

où "Valeur" est la notation de valeur du type parent.

37.1.2 Un ensemble de valeur "Valeur-Unique" est la valeur unique du type parent spécifiée par "Valeur". Cette notation peut être appliquée à tous les types parents.

#### 37.2 Sous-type contenu

37.2.1 La notation du "Sous-type-Contenu" est:

Sous-type-Contenu ::= INCLUDES Type

37.2.2 Un ensemble de valeurs "Sous-type-Contenu" est composé de toutes les valeurs du "Type", qui doit lui-même être un sous-type du type parent. Cette notation peut être appliquée à tous les types parents.

### 37.3 Plage de valeurs

37.3.1 La notation de "Plage-Valeurs" est:

Plage-Valeurs ::= Limite-Inférieure ..Limite-Supérieure

37.3.2 Un ensemble de valeurs "Plage-Valeurs" est composé de toutes les valeurs d'une plage qui est désignée en spécifiant les valeurs numériques de ses limites. Cette notation peut seulement être appliquée au type Entier, au type Réel et aux types dérivés de ces derniers par étiquetage ou sous-typage.

*Remarque* – Pour le sous-typage, "PLUS-INFINITY" est supérieur à toutes les valeurs "Réel-Numérique" et "MINUS-INFINITY" est inférieur à toutes les valeurs "Réel-Numérique".

37.3.3 Chaque limite de la plage est soit fermée (auquel cas cette limite est incluse dans l'ensemble de valeurs) soit ouverte (auquel cas la limite n'y est pas incluse). Quand l'extrémité est ouverte, sa spécification inclut un symbole inférieur-à (" $<$ "):

Limite-Inférieure ::= Limite-Inférieure | Limite-Inférieure  $<$

Limite-Supérieure ::= Limite-Supérieure |  $<$  Limite-supérieure

37.3.4 Une limite peut également ne pas être spécifiée, auquel cas la plage s'étend dans ce sens aussi loin que l'autorise le type parent:

Limite-Inférieure ::= Valeur | MIN

Limite-Supérieure ::= Valeur | MAX

### 37.4 Limitation de taille

37.4.1 La notation de "Limitation-Taille" est:

Limitation-Taille ::= SIZE Spécif-Sous-type

37.4.2 Une "Limitation-Taille" ne peut s'appliquer qu'aux types Chaîne binaire, Chaîne d'octets, Chaîne de caractères, Ensemble-de ou Séquence-de ou aux types formés par étiquetage ou sous-typage de ces derniers.

37.4.3 La "Spécif-Sous-type" spécifie les valeurs entières autorisées pour la longueur des nombres de l'ensemble des valeurs et a la forme de n'importe quelle spécification de sous-type qui peut être appliquée au type parent suivant:

INTEGER (0..MAX)

37.4.4 L'unité de mesure dépend comme suit du type parent:

Type	Unité de mesure
Chaîne binaire	bit
Chaîne d'octets	octet
Chaîne de caractères	caractère
Ensemble-de	valeur de composant
Séquence-de	valeur de composant

### 37.5 Alphabet autorisé

37.5.1 La notation d'"Alphabet" est:

Alphabet-Autorisé ::= FROM Spécif-Sous-type

37.5.2 Un ensemble de valeurs "Alphabet-Autorisé" est composé de toutes les valeurs qui peuvent être construites en utilisant un sous-alphabet de la chaîne parent. Cette notation ne s'applique qu'aux types Chaîne de caractères ou qui en dérivent par étiquetage ou sous-typage.

37.5.3 La "Spécif-Sous-type" spécifie les caractères qui peuvent apparaître dans la chaîne de caractères, et peut être n'importe quelle spécification de sous-type applicable au sous-type obtenu en appliquant la spécification de sous-type "SIZE(1)" au type parent.

## 37.6 *Sous-typage interne*

37.6.1 La notation des "Contraintes-Type-Interne" est:

Contraintes-Type-Interne ::=  
WITH COMPONENT Contrainte-Type-Unique |  
WITH COMPONENTS Contrainte-Types-Multiples

37.6.2 Des "Contraintes-Type-Interne" incluent seulement dans l'ensemble des valeurs celles qui satisfont un ensemble de contraintes portant sur la présence et/ou les valeurs des composants du type parent. Une valeur du type parent n'est incluse dans le sous-type que si elle satisfait toutes les contraintes explicites ou implicites (voir le § 37.6.6). Cette notation peut être appliquée aux types Ensemble-de, Séquence-de, Ensemble, Séquence et Choix, ainsi qu'aux types qui en dérivent par étiquetage ou sous-typage.

37.6.3 Une contrainte ayant la forme d'une spécification de valeur de sous-type est prévue pour les types qui sont définis en termes d'un unique autre type (interne) (Ensemble-de, Séquence-de et types qui en dérivent par étiquetage ou sous-typage). La notation de cette spécification est "Contrainte-Type-Unique".

Contrainte-Type-Unique ::= Spécif-Sous-type

La Spécif-Sous-type définit un sous-type de l'unique autre type (interne). Une valeur du type parent est un membre de l'ensemble des valeurs du sous-type si et seulement si chaque valeur interne appartient au sous-type obtenu en appliquant la "Spécif-Sous-type" au type interne.

37.6.4 Un certain nombre de contraintes s'appliquant aux types internes peuvent être définies: elles concernent les types qui sont définis en termes de plusieurs autres types (internes) (Choix, Ensemble, Séquence et types qui en dérivent par étiquetage ou sous-typage). La notation de ces contraintes est "Contraintes-Types-Multiples":

Contraintes-Types-Multiples ::=  
Spécification-Complète | Spécification-Partielle  
Spécification-Complète ::= {Contraintes-Types}  
Spécification-Partielle ::= {..., Contraintes-Types}  
Contraintes-Types ::=  
Contrainte-Nommée |  
Contrainte-Nommée, Contraintes-Types  
Contrainte-Nommée ::= Identificateur Contrainte | Contrainte

37.6.5 Les "Contraintes-Types" contiennent une liste de contraintes s'appliquant aux types composants du type parent. Pour un type séquence, les contraintes doivent figurer dans l'ordre de la séquence. Le type interne auquel les contraintes s'appliquent est identifié par son identificateur, s'il en a un, ou par sa position, dans le cas des types séquences.

*Remarque* – Quand le type interne n'a pas d'identificateur, la notation peut être ambiguë.

37.6.6 Les "Contraintes-Types-Multiples" comprennent soit une "Spécification-Complète" ou une "Spécification-Partielle". Si la "Spécification-Partielle" est utilisée, il existe une contrainte selon laquelle "ABSENT" figure implicitement pour tous les types internes qui ne sont pas explicitement énumérés (voir le § 37.6.8) et chaque type interne qui n'est pas marqué "OPTIONAL" ou "DEFAULT" dans le type parent, doit figurer explicitement. Quand la "Spécification-Partielle" est utilisée, il n'y a pas de contraintes implicites, et tout type interne peut être omis de la liste.

37.6.7 Les contraintes s'appliquant à un type interne particulier peuvent concerner sa présence (dans des valeurs du type parent), sa valeur ou les deux. La notation est "Contrainte":

Contrainte ::= Contrainte-Valeur Contrainte-Présence

37.6.8 Une contrainte concernant la valeur d'un type interne est exprimée par la notation "Contrainte-Valeur":

Contrainte-Valeur ::= Spécif-Sous-type | vide

La contrainte est satisfaite par une valeur du type parent, si et seulement si la valeur interne appartient au sous-type spécifié par la "Spécif-Sous-type" appliquée au type interne.

37.6.9 Une contrainte portant sur la présence d'un type interne est exprimée par la notation "Contrainte-Présence":

Contrainte-Présence ::= PRESENT | ABSENT | vide | OPTIONAL

La signification de ses possibilités, et les situations dans lesquelles elles sont autorisées, sont définies aux § 37.6.9.1 à 37.6.9.3.

37.6.9.1 Si le type parent est une séquence ou un ensemble, un type élément marqué "OPTIONAL" peut être contraint à être "PRÉSENT", (auquel cas la contrainte est satisfaite si et seulement si la valeur d'élément correspondante figure) ou "ABSENT" (auquel cas la contrainte est satisfaite si et seulement si la valeur d'élément correspondante ne figure pas) ou "OPTIONAL" (auquel cas aucune contrainte n'est placée sur la présence de la valeur d'élément correspondante).

37.6.9.2 Si le type parent est un choix, un type composant peut être contraint à être "ABSENT", auquel cas la contrainte est satisfaite si et seulement si le type composant correspondant n'est pas utilisé dans la valeur.

37.6.9.3 La signification d'une "Contrainte-Présence" vide dépend de la spécification employée: "Spécification-Complète" ou "Spécification-Partielle":

- a) dans une "Spécification-Complète", elle est équivalente à la contrainte d'être "PRÉSENT";
- b) dans une "Spécification-Partielle", aucune contrainte n'est imposée.

## ANNEXE A

(à la Recommandation X.208)

### La notation de macro

#### A.1 Introduction

Un mécanisme est prévu dans la notation ASN.1 pour permettre à son utilisateur de définir une nouvelle notation avec laquelle il peut construire et dénoter des types ASN.1 ou spécifier des valeurs de types. La nouvelle notation se définit en utilisant la notation "Définition-Macro". Une "Définition-Macro" spécifie simultanément une nouvelle notation de construction et de dénotation de type et une nouvelle notation de spécification de valeur. (Voir au § I.3 une illustration de l'emploi de la notation de macro.)

Avec une "Définition-Macro", l'utilisateur d'ASN.1 spécifie la nouvelle notation au moyen d'un ensemble de productions, d'une façon similaire à celle de la présente Recommandation. L'auteur de la définition de macro:

- a) spécifie la syntaxe complète à utiliser pour définir tous les types construits par la macro; (cette spécification de syntaxe est appelée à des fins d'analyse syntaxique, à chaque occurrence du nom de la macro dans la notation de type ASN.1);
- b) spécifie la syntaxe complète à utiliser pour une valeur d'un de ces types (cette spécification de syntaxe est appelée à des fins d'analyse syntaxique, chaque fois qu'une valeur du type macro est prévue);
- c) spécifie, comme valeur d'un type standard ASN.1 (de complexité quelconque), le type et la valeur résultants, pour toutes les occurrences de la notation de la valeur de la macro.

Une occurrence de la syntaxe définie par la définition de la macro peut contenir des occurrences de type ou de valeur (utilisant la notation standard ASN.1). Ces types ou valeurs (qui apparaissent dans l'utilisation de la notation de la macro) peuvent être associés, pour la durée de l'analyse syntaxique, à une référence locale de type, ou à une dénotation locale de type, ou à une dénotation locale de valeur, par des instructions appropriées figurant dans la définition de la macro. Il est également possible d'inclure, dans la définition de la macro, des affectations de type ASN.1. Ces affectations deviennent actives lorsque la catégorie syntaxique associée est appariée à un ou des items de l'occurrence de la nouvelle notation en cours d'analyse. Leur durée de vie est limitée à celle de l'analyse.

Lors de l'analyse d'une valeur de la nouvelle notation, les affectations faites pendant l'analyse de la notation du type correspondant sont disponibles. Une telle analyse est considérée comme précédant logiquement l'analyse de chaque occurrence de la notation de valeur.

Le type et la valeur résultant d'une utilisation de la nouvelle notation de valeur sont déterminés par la valeur (et le type de cette valeur) qui sont finalement affectés à la référence locale de valeur distincte identifiée par le mot clé VALUE, suivant le traitement de la définition de macro pour la nouvelle notation de type, suivie par celle de la nouvelle notation de valeur.

Chaque "Définition-Macro" définit une notation (une syntaxe) de définition du type, et une notation (une syntaxe) de définition de la valeur. Le type ASN.1 qui est défini par une occurrence de la nouvelle notation de type peut, mais pas nécessairement, dépendre de l'occurrence de la notation de valeur à laquelle le type est associé. De ce point de vue, l'utilisation de la nouvelle notation de type est semblable à un CHOICE: l'étiquette est indéterminée. Ainsi, dans ce cas, la nouvelle notation ne peut pas être utilisée quand une étiquette connue est nécessaire, et ne peut pas non plus être étiquetée explicitement.

## A.2 *Extensions aux items et au jeu de caractères ASN.1*

Les caractères | et > sont utilisés dans la notation des macros.

Les items spécifiés dans les paragraphes suivants sont également utilisés.

### A.2.1 *Dénotation-macro*

Nom de l'item: dénotation-macro

Une "dénotation-macro" est composée de la séquence de caractères spécifiée pour une "dénotation-type" au § 8.2, sauf que tous les caractères doivent être majuscules. Dans un module donné, la même séquence de caractères ne doit pas être utilisée à la fois pour une dénotation-type et une dénotation-macro.

### A.2.2 *Dénotation-production*

Nom de l'item: dénotation-production

Une "dénotation-production" est composée des séquences de caractères spécifiées pour une "dénotation-type" au § 8.2.

### A.2.3 *Dénotation-type-local*

Nom de l'item: dénotation-type-local

Une "dénotation-type-local" est composée de la séquence de caractères spécifiée pour une "dénotation-type" au § 8.2. Une "dénotation-type-local" est utilisée comme identificateur pour les types qui sont reconnus lors de l'analyse syntaxique d'une occurrence de la nouvelle notation de type ou de valeur.

### A.2.4 *Dénotation-valeur-locale*

Nom de l'item: dénotation-valeur-locale

Une "dénotation-valeur-locale" est composée de la séquence de caractères spécifiée pour une "dénotation-valeur" au § 8.2. Une "dénotation-valeur-locale" est utilisée comme identificateur pour les valeurs qui sont reconnues lors de l'analyse syntaxique d'une occurrence de la nouvelle notation de type ou de valeur.

### A.2.5 *Item de sélection*

Nom de l'item: " | "

Cet item est composé du seul caractère |.

### A.2.6 *Item de terminaison de définition*

Nom de l'item: >

Cet item est composé du seul caractère >.

*Remarque* – L'item < de début de définition est défini au § 8.13.

### A.2.7 *Item terminal syntaxique*

Nom de l'item: chaîne-a

Une "chaîne-a" est composée d'un nombre quelconque (éventuellement zéro) de caractères du jeu de caractères ASN.1 (voir le § 7), encadrés par des ". Dans une "chaîne-a" le caractère " sera représenté par un couple de ".

Remarque – L'emploi de "chaîne-a" dans la notation de macro indique l'apparition, au point correspondant de la syntaxe en cours d'analyse, des caractères entre guillemets (").

TABLEAU 8/X.208

Séquences spécifiées par des items

Nom de l'item	Définie au §
"string"	toute séquence de caractères
"identifiant"	8.3 – Identificateurs
"number"	8.8 – Nombres
"empty"	8.7 – Vide

A.2.8 Items mot clé de catégorie syntaxique

Nom des items: "string"  
 "identifiant"  
 "number"  
 "empty"

Les items des noms ci-dessus doivent être composés (en notation de macro) des séquences de caractères du nom, à l'exclusion du caractère guillemet ("). Ces items sont utilisés dans la notation de macro pour spécifier l'occurrence, dans une utilisation de la nouvelle notation, de certaines séquences de caractères. Le tableau 8/X.208 indique les séquences de caractères apparaissant dans la nouvelle notation et spécifiées par un item, en indiquant le paragraphe de la présente notation qui définit cette séquence de caractères.

Remarque – La notation de macro ne fait pas de différence entre les identificateurs et les dénominations commençant par une lettre majuscule ou minuscule. Ceci pour des raisons historiques.

A.2.9 Autres items mot clé

Nom des items: MACRO  
 TYPE  
 NOTATION  
 VALUE  
 value  
 type

Les items des noms ci-dessus sont composés de la séquence de caractères du nom.

Les items spécifiés dans les § A.2.2 à A.2.4 inclus, ne doivent pas être une des séquences du § A.2.9, sauf lorsqu'ils sont utilisés comme spécifié ci-après.

Le mot clé "MACRO" doit être utilisé pour commencer une définition de macro. Le mot clé "TYPE NOTATION" doit être utilisé comme nom de la production qui définit la nouvelle notation. Le mot clé "VALUE NOTATION" doit être utilisé comme nom de la production qui définit la nouvelle notation de valeur. Le mot clé "VALUE" doit être utilisé comme "dénomination-valeur-locale" à laquelle la valeur résultante est affectée. Le mot clé "value" doit être utilisé pour spécifier que chaque occurrence de la nouvelle notation contient, à ce point, en notation ASN.1, une certaine valeur d'un "type" (spécifié dans la définition de macro). Le mot clé "type" doit être utilisé pour spécifier que chaque occurrence de la nouvelle notation contient, à ce point, en notation ASN.1, un certain "Type".

### A.3 Notation de définition de macro

#### A.3.1 Une macro doit être définie par la notation "Définition-Macro":

Définition-Macro ::=

dénotation-macro  
MACRO  
"::="  
Substance-Macro

Substance-Macro ::=

BEGIN Corps-Macro END |  
dénotation-macro |  
Dénotation-macro-externe

Corps-Macro ::=

Production-Type  
Production-Valeur  
Productions-Support

Production-Type ::=

TYPE-NOTATION  
"::="  
Liste-Macros-Possibles

Production-Valeur ::=

VALUE NOTATION  
"::="  
Liste-Macros-Possibles

Productions-Support ::=

Liste-Productions |  
vide

Liste-Productions ::=

Production |  
Liste-Productions Production

Production ::=

dénotation-production  
"::="  
Liste-Macros-Possibles

Dénotation-macro-externe ::=

dénotation-module . dénotation-macro

*Remarque* – Il est convenu qu'une seule définition macro peut servir à dénoter d'autres macros, à condition que ce soit pour un complément d'étude.

A.3.2 Si la possibilité "dénotation-macro" de "Substance-Macro" est choisie, le module contenant la définition de macro doit:

- a) soit contenir une autre définition de macro définissant cette "dénotation-macro";
- b) soit contenir la "dénotation-macro" dans ses "Symboles-Importés".

A.3.3 Si la possibilité "Dénotation-macro-externe" de "Substance-Macro" est choisie, le module dénoté par "Dénotation-Module" doit contenir une définition de macro définissant la "dénotation-macro". La définition associée est alors également associée à la "dénotation-macro" en cours de définition.

A.3.4 La chaîne de définitions qui peut résulter de l'application répétée des règles des § A.3.2 à A.3.3 se termine par une "Définition-Macro" qui utilise la possibilité "BEGIN Corps-Macro END" et c'est ce "Corps-Macro" qui définit le type et la notation de la valeur pour la macro en cours de définition.

A.3.5 Chaque "dénotation-production" qui se présente dans une "Déf-Symbole" (voir § A.3.9) doit se présenter une fois et une seule comme premier élément d'une "Production".

A.3.6 Chaque occurrence de la nouvelle notation de type doit commencer par la séquence de caractères de la "dénotation-macro" suivie de l'une des séquences de caractères dénotée par "TYPE NOTATION" après application des productions spécifiées dans la définition de la macro.

A.3.7 Chaque occurrence de la nouvelle notation de valeur doit être composée d'une des séquences de caractères dénotée par "VALUE NOTATION", après application des productions spécifiées dans la définition de la macro.

A.3.8 La "Liste-Macros-Possibles" d'une production spécifie les ensembles de séquences de caractères possibles dénotés par cette production. Elle est spécifiée par:

```
Liste-Macros-Possibles ::=
    Macro-Possible |
    Liste-Macros-Possibles " | " Macro-Possible
```

L'ensemble des séquences de caractères dénotées par la "Liste-Macros-Possibles" est composé de toutes les séquences de caractères qui sont dénotées par une des productions "Macros-Possibles" de la "Liste-Macros-Possibles"

A.3.9 La notation d'une "Macro-possible" est:

```
Macro-possible ::= Liste-Symboles
Liste-Symboles ::=
    Elément-Symbole |
    Liste-Symboles Elément-Symbole
Elément-Symbole ::=
    Déf-Symbole |
    Définitions-Incluses
Déf-Symbole ::=
    chaîne-a |
    dénotation-production |
    "chaîne" |
    "identificateur" |
    "nombre" |
    "vide" |
    type |
    type (dénotation-type-local) |
    valeur (Type-Macro) |
    valeur (dénotation-valeur-locale Type-Macro) |
    valeur (VALUE Type-Macro)
Type-Macro ::= dénotation-type-local
Type
```

*Remarque* – Dans une macro, un "Type-Macro" qui y est défini peut figurer en n'importe quel point auquel ASN.1 spécifie un "Type".

Une "Macro-possible" dénote toutes les chaînes de caractères qui sont formées en prenant n'importe laquelle des chaînes de caractères dénotées par la première "Déf-Symbole" de la "Liste-Symboles", suivies d'une quelconque des chaînes de caractères dénotées par la deuxième "Déf-Symbole" de la "Liste-Symboles", et ainsi de suite jusques et y compris la dernière "Déf-Symbole" de la "Liste-Symboles".

*Remarque* – Les "Définitions-Incluses" (le cas échéant) ne jouent aucun rôle direct dans la détermination de ces chaînes.

A.3.10 Une "chaîne-a" dénote la séquence de caractères contenue dans la "chaîne-a", sans la paire de " " qui l'encadre.

A.3.11 Une "dénotation-production" dénote toute séquence de caractères spécifiée par la "Production" qu'elle identifie.

A.3.12 Les séquences de caractères dénotées par les quatre possibilités suivantes de "Déf-Symbole" sont spécifiées dans le tableau 8/X.208.

*Remarque* – Les séquences de caractères dénotées par la "chaîne" doivent se terminer dans une occurrence de la notation de macro, par l'apparition d'une séquence dénotée par la "Déf-Symbole" suivante de la "Liste-Symboles".

A.3.13 Un "Type" dénote toute séquence de symboles qui forme une notation "Type" comme spécifié au § 12.1.

*Remarque* – Le "Type-Défini" du § 12.1 peut dans ce cas contenir une "dénotation-type-local" dénotant un type défini dans la notation de macro.

A.3.14 Une "type(dénotation-type-local)" dénote toute séquence de symboles qui forme un "Type" comme spécifié au § 12.1, mais affecte en plus ce type à la "dénotation-type-local". Cette "dénotation-type-local" peut faire l'objet d'une affectation ultérieure.

A.3.15 Une "value(Type-Macro)" dénote toute séquence de symboles qui forme une notation "Value" (comme spécifié au § 12.7) pour le type spécifié par "Type-Macro".

A.3.16 Une "value(dénotation-valeur-locale Type-Macro)" dénote toute séquence de symboles qui forme une notation "Valeur" (comme spécifié au § 12.7) pour le type spécifié par "Type-Macro", mais affecte en plus la valeur spécifiée par la notation de valeur à la "dénotation-valeur-locale". Cette "dénotation-valeur-locale" peut faire l'objet d'une affectation ultérieure.

A.3.17 Une "value(VALUE Type-Macro)" dénote toute séquence de symboles qui forme une notation "Value" (comme spécifié au § 12.7) pour le type spécifié par "Type-Macro", mais en plus, renvoie la valeur en tant que valeur spécifiée par la notation de valeur. Le type de la valeur renvoyée est le type dénoté par "Type-Macro".

A.3.18 Lors de l'analyse de toute occurrence de la nouvelle notation de valeur, VALUE reçoit une et une seule affectation (comme spécifié au § A.3.17 ou au § A.3.19).

A.3.19 La notation des "Définition-Incluses" est:

```
Définitions-Incluses ::=
    < Liste-Définitions-Incluses >

Liste-Définitions-Incluses ::=
    Définition-Incluse |
    Liste-Définitions-Incluses
    Définition-Incluse

Définition-Incluse ::=
    Affectation-Type-Local |
    Affectation-Valeur-Locale

Affectation-Type-Local ::=
    dénotation-type-local
    "::="
    Type-Macro

Affectation-Valeur-Local ::=
    dénotation-valeur-locale
    Type-Macro
    "::="
    Valeur-Macro

Valeur-Macro ::=
    Valeur |
    dénotation-valeur-locale
```

L'affectation d'un "Type-Macro" à une "dénotation-type-local" (ou d'une "Valeur-Macro" à une "dénotation-valeur-locale") dans une "Définitions-Incluses" prend effet durant l'analyse syntaxique d'une occurrence de la nouvelle notation, au moment où les "Définitions-Incluses" sont rencontrées, et reste en vigueur jusqu'à la redéfinition de la "dénotation-type-local" ou de la "dénotation-valeur-locale".

*Remarque 1* – L'utilisation de la "dénotation-type-local" ou de la "dénotation-valeur-locale", ailleurs dans la "Macro-Possible", implique des hypothèses sur la nature de l'algorithme d'analyse syntaxique. Ces hypothèses devraient être précisées par un commentaire. Par exemple, l'utilisation de la "dénotation-type-local" suivant textuellement "Définitions-Incluses" implique une analyse syntaxique de gauche à droite.

*Remarque 2* – La "dénotation-valeur-locale" "VALUE" peut se voir affecter une valeur soit par la structure "value(VALUE Type-Macro)" soit par une "Définitions-Incluses". Dans les deux cas, la valeur est renvoyée, comme spécifié au § A.3.17.

#### A.4 Utilisation de la nouvelle notation

Chaque fois qu'une notation de "Type" (ou de "Valeur") est exigée par la présente Recommandation, une occurrence de la notation de type (ou de la notation de valeur) définie par une macro dans le même module, peut être utilisée, à condition que la macro soit:

- a) définie dans le même module;
- b) importée dans le module du fait de la présence de la "dénotation-macro" dans les "Symboles-Importés" du module.

Pour permettre la dernière possibilité, une "dénotation-macro" peut apparaître comme un "Symbole" au § 9.1.

*Remarque 1* – Cette extension à la notation standard ASN.1 n'apparaît pas dans le corps de la présente Recommandation.

*Remarque 2* – Il est possible de construire des modules comprenant des séquences d'affectation de type et des définitions de macro, quelle que soit la complexité impliquée de l'analyse syntaxique des valeurs DEFAULT.

## ANNEXE B

(à la Recommandation X.208)

### Affectation par l'ISO des valeurs des composants de l'identificateur d'objet

B.1 Trois arcs sont spécifiés à partir d'une racine. L'affectation des valeurs et identificateurs, ainsi que l'autorité d'affectation des valeurs des composants des niveaux suivants sont stipulées comme suit:

Valeur	Identificateur	Autorité
0	ccitt	CCITT
1	iso	ISO
2	joint-iso-ccitt	Voir annexe D

*Remarque* – La suite de cette annexe ne concerne que l'affectation de valeurs par l'ISO.

B.2 Les identificateurs "ccitt", "iso" et "joint-iso-ccitt", affectés ci-dessus, peuvent chacun être utilisés comme une "Forme-Nom".

B.3 Quatre arcs sont spécifiés à partir du noeud identifié par "iso". L'affectation des valeurs et des identificateurs est stipulée comme suit:

Valeur	Identificateur	Autorité
0	norme	Voir B.4
1	autorité-enregistrement	Voir B.5
2	comité-membre	Voir B.6
3	organisation-identifiée	Voir B.7

Ces identificateurs peuvent être utilisés comme une "Forme-Nom".

B.4 Les arcs issus de "norme" ont chacun la valeur du numéro d'une norme internationale. Si la norme internationale est en plusieurs parties, un arc additionnel doit correspondre au numéro de partie, sauf si ceci est spécifiquement exclu dans le texte de la norme internationale. Les valeurs des autres arcs seront telles que définies dans la présente norme internationale.

*Remarque* – Si une norme internationale en une seule partie affecte des identificateurs d'objet, et devient ensuite une norme internationale en plusieurs parties, les affectations stipulées par cette norme doivent continuer de se faire comme si elle était en une seule partie.

B.5 Les arcs issus de l'"autorité d'enregistrement" sont réservés à un additif à cette norme internationale, qui sera élaboré parallèlement à la définition des procédures d'identification des autorités d'enregistrement OSI spécifiques.

B.6 Les arcs issus de "comité membre" ont les valeurs du code numérique de pays à trois chiffres, spécifiées dans la norme ISO 3166, qui identifie le comité membre ISO de ce pays (voir remarque). La "Forme-Nom" du composant de l'identificateur d'objet n'est pas autorisée avec ces identificateurs. La présente Recommandation ne définit pas d'arc issu du "code-pays".

*Remarque* – L'existence d'un code de pays dans la norme ISO 3166 n'implique pas nécessairement qu'il existe un comité membre ISO représentant ce pays, ou que le comité membre ISO de ce pays affecte des composants d'identificateurs d'objet.

B.7 Les arcs issus de "organisation-identifiée" ont des valeurs définies par un prescripteur de code international (ICD), désigné par l'autorité d'enregistrement de la norme ISO 6523, qui identifie une organisation originaire spécifiquement enregistrée par cette autorité comme affectant des composants d'identificateurs d'objet (voir remarques 1 et 2). Les arcs issus d'ICD ont des valeurs de "code d'organisation" affectées par l'organisation originaire, conformément à la norme ISO 6523. Les arcs issus du "code d'organisation" ne sont pas définis par la présente Recommandation (voir remarque 3).

*Remarque 1* – Le fait de demander que les organisations originaires soient enregistrées par l'autorité d'enregistrement de la norme ISO 6523, en tant qu'organisation affectant des codes pour des composants d'identificateurs d'objet, garantit que seules sont attribuées des valeurs numériques conformes à la présente Recommandation.

*Remarque 2* – Le fait de déclarer qu'une organisation originaire affecte des codes d'organisation à des composants d'identificateurs d'objet n'interdit pas d'utiliser ces codes à d'autres fins.

*Remarque 3* – Il est supposé que les organisations identifiées par le "code d'organisation" définiront d'autres arcs d'une façon qui garantira l'affectation de valeurs uniques.

*Remarque 4* – Les dispositions stipulées au § B.7 ont pour effet que toute organisation peut obtenir un code d'organisation auprès d'une organisation originaire appropriée, et peut ensuite affecter des valeurs d'identificateurs d'objet pour son propre usage, avec l'assurance que ces valeurs n'entreront pas en conflit avec celles affectées par d'autres organisations. Ainsi, un industriel pourrait par exemple affecter un identificateur d'objet à des formats d'informations propres à son entreprise.

## ANNEXE C

(à la Recommandation X.208)

### Affectation par le CCITT des valeurs des composants de l'identificateur d'objet

C.1 Trois arcs sont spécifiés à partir d'une racine. L'affectation des valeurs et identificateurs, ainsi que l'autorité d'affectation des valeurs des composants des niveaux suivants sont stipulées comme suit:

Valeur	Identificateur	Autorité
0	ccitt	CCITT
1	iso	ISO
2	joint-iso-ccitt	Voir annexe D

*Remarque* – La suite de cette annexe ne concerne que l'affectation de valeurs par l'ISO.

C.2 Les identificateurs "ccitt", "iso" et "joint-iso-ccitt", affectés ci-dessus, peuvent chacun être utilisés comme une "Forme-Nom".

C.3 Quatre arcs sont spécifiés à partir du noeud identifié par "ccitt". L'affectation des valeurs et des identificateurs est stipulée comme suit:

Valeur	Identificateur	Autorité
0	recommandation	Voir § C.4
1	question	Voir § C.5
2	administration	Voir § C.6
3	opérateur-réseau	Voir § C.7

Ces identificateurs peuvent être utilisés comme une "Forme-Nom".

C.4 Les arcs issus de "recommandation" ont une valeur de 1 à 26, l'identificateur affecté étant une lettre de a à z. Les arcs issus de ceux-ci ont les numéros des Recommandations du CCITT des séries identifiées par la lettre. Les arcs qui en sont issus sont déterminés comme jugé utile par la Recommandation du CCITT. Les identificateurs a à z peuvent être utilisés comme une Forme-Nom.

C.5 Les arcs issus de "question" ont des valeurs correspondant aux Commissions d'études du CCITT, qualifiées par la période d'études. La valeur est donnée par la formule:

$$\text{numéro de la Commission d'études} + (\text{période} * 32)$$

où "période" a la valeur 0 pour 1984-1988, 1 pour 1988-1992 et ainsi de suite, le multiplicateur 32 étant décimal.

Les arcs issus de chaque Commission d'études ont les valeurs correspondant aux questions inscrites au programme de travail de ladite Commission, les arcs du niveau suivant sont déterminés comme jugé utile par le groupe auquel est confiée l'étude de la question (par exemple, groupe de travail ou groupe à rapporteur).

C.6 Les arcs issus de l'"Administration" ont les valeurs des IPD de X.121. Les arcs du niveau suivant sont déterminés comme jugé utile par l'Administration du pays identifié par l'IPD de X.121.

C.7 Les arcs issus de l'"opérateur-réseau" ont les valeurs des DNIC de X.121. Les arcs du niveau suivant sont déterminés comme jugé utile par l'Administration (ou l'EPR identifiée par le CIRD).

## ANNEXE D

(à la Recommandation X.208)

### Affectation commune des valeurs des composants de l'identificateur d'objet

D.1 Trois arcs sont spécifiés à partir d'une racine. L'affectation des valeurs et identificateurs, ainsi que l'autorité d'affectation des valeurs des composants des niveaux suivants sont stipulées comme suit:

Valeur	Identificateur	Autorité
0	ccitt	CCITT
1	iso	ISO
2	joint-iso-ccitt	Voir annexe D

*Remarque* – La suite de cette annexe ne concerne que l'affectation de valeurs communes ISO-CCITT.

D.2 Les identificateurs "ccitt", "iso" et "joint-iso-ccitt", affectés ci-dessus, peuvent chacun être utilisés comme une "Forme-Nom".

D.3 Les arcs issus de "joint-iso-ccitt" ont des valeurs qui sont affectées et agréées périodiquement par l'ISO et le CCITT, pour identifier les domaines d'activités de normalisation communes au CCITT et à l'ISO, conformément aux "Procédures d'affectation des valeurs des composants de l'identificateur d'objet pour utilisation commune par l'ISO et le CCITT"<sup>1)</sup>.

D.4 Les arcs issus de chaque arc déterminé par les mécanismes décrits au § D.3 doivent être affectés conformément aux mécanismes institués lors de l'affectation de cet arc.

*Remarque* – Il est prévu que ceci fera l'objet d'une délégation d'autorité, permettant l'acceptation commune par les Rapporteurs CCITT et ISO dans les domaines de travail communs.

D.5 Les premières normes ISO et Recommandations du CCITT relevant du domaine d'activité commun à ces organisations demandent l'attribution d'identificateur d'objet préalablement à l'institution des procédures du § D.3 et procèdent donc à des attributions conformément aux annexes B ou C. Les objets informationnels identifiés de cette manière par des normes ISO ou par des Recommandations du CCITT, ne devront pas modifier leurs identificateurs d'objet avant que les procédures décrites au § D.3 ne soient instituées.

---

<sup>1)</sup> L'Autorité d'enregistrement chargée de l'affectation des valeurs des composants de l'identificateur d'objet, pour des utilisations communes par l'ISO et le CCITT est l'ANSI (American National Standards Institute), 1430 Broadway, New York, NY 1008, Etats-Unis d'Amérique

## APPENDICE I

(à la Recommandation X.208)

### Exemples et directives

Cet appendice présente des exemples d'utilisation d'ASN.1 pour la description de structures de données (fictives). Il propose également des conseils ou directives concernant l'utilisation des diverses caractéristiques d'ASN.1.

#### I.1 Exemple d'un "enregistrement salarié"

L'utilisation d'ASN.1 est illustrée par un enregistrement simple concernant un salarié fictif.

##### I.1.1 Description informelle de l'enregistrement "salarié"

La structure de l'enregistrement "salarié" est décrite ci-après, avec sa valeur pour un individu particulier.

Nom:	John P. Smith
Fonction:	Directeur
Matricule:	51
Date d'engagement:	17 septembre 1971
Nom du conjoint:	Mary T. Smith
Nombre d'enfants:	2

##### Renseignements sur enfant

Nom:	Ralph T. Smith
Date de naissance:	11 novembre 1957

##### Renseignement sur enfant

Nom:	Susan B. Jones
Date de naissance:	17 juillet 1959

##### I.1.2 Description ASN.1 de la structure de l'enregistrement

La structure de chaque enregistrement "salarié" est décrite formellement en utilisant la notation standard des types de données.

Enregistrement-Salarié ::= [APPLICATION 0] IMPLICIT SET

{	Nom
fonction	[0] Chaîne-Visible
matricule	Matricule-Salarié
Date-engagement	[1] Date
Nom-conjoint	[2] Nom
Enfants	[3] IMPLICIT SEQUENCE OF Renseignements-Enfant
	DEFAULT {} }

Renseignements-Enfants ::= SET

{	Nom
date-Naissance	[0] Date }

Nom ::= [APPLICATION 1] IMPLICIT SEQUENCE

{Prénom	Chaîne-Visible
initiale	Chaîne-Visible
nom-Famille	Chaîne-Visible }

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

Date ::= [APPLICATION 3] IMPLICIT Chaîne-Visible-- YYYYMMDD

Cet exemple illustre un aspect de la puissance d'analyse de la syntaxe ASN.1. La structure syntaxique "DEFAULT" ne peut être appliquée qu'à un élément de "SEQUENCE" ou de "SET"; elle ne peut pas être appliquée à

un élément de "SEQUENCE OF". Ainsi, "DEFAULT {}" de "Enregistrement-Salarié" s'applique à "enfants" et non à "Renseignements-Enfant".

### I.1.3 Description ASN.1 d'une valeur de l'enregistrement

La valeur de l'enregistrement "salarié" de John Smith est décrite formellement ci-dessous en utilisant la notation standard des valeurs de données.

```
{
    {prénom "John",initiale "P",nom-Famille "Smith"}
fonction                "Directeur"
matricule                51
date-Engagement        "19710917"
nom-Conjoint           {prénom "Mary",initiale "T",nom-Famille "Smith"}
enfants
    {{{prénom "Ralph",initiale "T",nom-Famille "Smith"}
      date-Naissance "19571111"}
     {{{prénom "Susan",initiale "B",nom-Famille "Jones"}
      date-Naissance "19590717"}}}}
```

## I.2 Directives pour l'utilisation de la notation

La souplesse des types de données et de la notation formelle définis par la présente Recommandation permet la conception d'une très large gamme de protocoles. Toutefois, cette souplesse peut parfois porter à confusion, en particulier lors d'une première utilisation de la notation. Cet appendice vise à réduire au minimum les risques de confusion, en proposant des directives et des exemples concernant l'utilisation de la notation. Pour chacun des types de données prédéfinis, une ou plusieurs directives d'emploi sont proposées. Les types chaînes de caractères (par exemple Chaîne-Visible) et les types définis dans la section 3 ne sont pas traités ici.

### I.2.1 Booléen

I.2.1.1 Utiliser un type Booléen pour représenter les valeurs d'une variable logique (c'est-à-dire à deux états), par exemple la réponse affirmative ou négative à une question.

*Exemple:*

```
Salarié ::= BOOLEAN
```

I.2.1.2 Pour affecter un nom de référence à un type Booléen, en choisir un qui décrive l'état vrai.

*Exemple:*

```
Marié ::= BOOLEAN
```

et non

```
Situation-Famille ::= BOOLEAN
```

Voir également le § I.2.3.2

### I.2.2 Entier

I.2.2.1 Utiliser un type Entier pour représenter les valeurs (dont la grandeur est, en pratique, illimitée) d'une variable cardinale ou entière.

*Exemple:*

```
Solde-Compte-Bancaire ::= INTEGER
--en centimes, négatif signifie découvert
```

I.2.2.2 Définir comme valeur distinguée, les valeurs minimale et maximale autorisées d'un Entier.

*Exemple:*

```
Quantième-Mois ::= INTEGER {premier(1), dernier(31)}
```

### I.2.3 Enuméré

I.2.3.1 Utiliser un type Enuméré avec des valeurs distinguées pour représenter les valeurs d'une variable qui peut avoir trois états ou plus. Attribuer des valeurs en partant de zéro si leur seule contrainte est la distinction.

*Exemple:*

```
Jour-Semaine ::= ENUMERATED
  { dimanche(0), lundi(1), mardi(2), mercredi(3), jeudi(4), vendredi(5), samedi(6) }
```

I.2.3.2 Utiliser un type Enuméré pour représenter les valeurs d'une variable qui ne peut avoir que deux états pour l'instant, mais qui pourra en avoir d'autres dans une version ultérieure du protocole.

*Exemple:*

```
Situation-Famille ::= ENUMERATED { célibataire(0), marié(1) }
```

en prévision de

```
Situation-Famille ::= ENUMERATED { célibataire(0), marié(1), veuf(2) }
```

### I.2.4 Réel

I.2.4.1 Utiliser un type Réel pour représenter un nombre approximatif.

*Exemple:*

```
Angle-En-Radians ::= REAL
pi REAL ::= { 3141592653589793238462643383279, 10, -30 }
```

### I.2.5 Chaîne binaire

I.2.5.1 Utiliser une Chaîne binaire pour représenter des données binaires dont le format et la longueur ne sont pas spécifiés, ou spécifiés ailleurs, et dont la longueur en bits n'est pas nécessairement un multiple de huit.

*Exemple:*

```
Page-Télécopie-G3 ::= BIT STRING
  -- séquence binaire conforme à la
  -- Recommandation T.4. du CCITT
```

I.2.5.2 Définir le premier et le dernier bit significatif d'une Chaîne binaire à longueur fixée comme bit distingué.

*Exemple:*

```
Franchise ::= BIT STRING { premier(0), dernier(3) }
```

I.2.5.3 Utiliser une Chaîne binaire pour représenter les valeurs d'une **table binaire**, c'est-à-dire d'un ensemble ordonné de variables logiques indiquant si une certaine condition est remplie pour chacun des objets d'un ensemble ordonné correspondant.

*Exemple:*

```
Jours-Ensoleillés-Du-Mois ::= BIT STRING { premier(1), dernier(31) }
  -- le Jour i était ensoleillé si et seulement si le bit 1 est égal à un
```

I.2.5.4 Utiliser une Chaîne binaire avec des valeurs remarquables, pour représenter les valeurs d'un ensemble de variables logiques en relation.

*Exemple:*

```
Statut-Personnel ::= BIT STRING
  { marié(0), salarié(1), ancien salarié(2), diplômé(3) }
```

## I.2.6 Chaîne d'octets

I.2.6.1 Utiliser un type Chaîne d'octets pour représenter des données binaires dont le format et la longueur ne sont pas spécifiés, ou sont spécifiés ailleurs, et dont la longueur en bits est un multiple de huit.

*Exemple:*

```
Image-Télécopie-G4 ::= OCTET STRING
-- séquence d'octets conforme aux
-- Recommandations du CCITT T.5 et T.6
```

I.2.6.2 Utiliser un type chaîne de caractères de préférence à un type Chaîne d'octets, quand un type approprié est disponible.

*Exemple:*

```
Prénom ::= Chaîne-Imprimable
```

I.2.6.3 Utiliser un type Chaîne d'octets pour représenter toute chaîne d'informations qui ne peut pas être représentée avec l'un des types chaîne de caractères. Ne pas oublier de spécifier le répertoire des caractères, avec leur codage en octets.

*Exemple:*

```
Chaîne-DCB-Compactée ::= OCTET STRING
-- chiffres 0 à 9 avec deux chiffres par octet,
-- chaque chiffre est codé de 0000 à 1001,
-- 11112 est utilisé comme élément de remplissage
```

## I.2.7 Vide

Utiliser un type Vide pour indiquer l'absence effective d'un élément d'une Séquence.

*Exemple:*

```
Identificateur-Patient ::= SEQUENCE
{ nom          Chaîne-Visible
  numéro-Chambre CHOICE
    { INTEGER
      NULL -- si patient non résident --}
}
```

*Remarque* – L'utilisation d'OPTIONAL donne le même résultat.

## I.2.8 Séquence et Séquence-de

I.2.8.1 Utiliser un type Séquence-de pour représenter un ensemble de variables dont les types sont identiques, le nombre élevé ou imprévisible et dont l'ordre est significatif.

*Exemple:*

```
Nom-Pays-Membre ::= SEQUENCE OF Chaîne-Visible
-- par ordre d'adhésion
```

I.2.8.2 Utiliser un type Séquence pour représenter un ensemble de variables dont les types sont les mêmes, dont le nombre est petit et connu et dont l'ordre est significatif, sous réserve que la composition de l'ensemble ait peu de chance de changer d'une version du protocole à la suivante.

*Exemple:*

```
Nom-Membres-Direction ::= SEQUENCE
{ président      Chaîne-Visible
  vice-Président Chaîne-Visible
  secrétaire     Chaîne-Visible }
```

I.2.8.3 Utiliser un type Séquence pour représenter un ensemble de variables de types différents, dont le nombre est petit, connu et l'ordre significatif, à condition que la composition de cet ensemble ait peu de chance de changer d'une version du protocole à la suivante.

*Exemple:*

```
Justificatifs ::= SEQUENCE
{ nom-utilisateur          Chaîne-Visible
  mot-passe                Chaîne-Visible
  numéro-compte            INTEGER }
```

I.2.8.4 Si les éléments d'un type Séquence sont en nombre fixe, mais les types différents, un nom de référence doit être affecté à chaque élément dont l'usage n'est pas entièrement évident d'après son type.

*Exemple:*

```
Fichier ::= SEQUENCE
{
  autre                    Type-Contenu
  contenu                  Attributs-Fichier
                          ANY }
```

Voir également les § I.2.5.3, I.2.5.4 et I.2.7.

## I.2.9 Ensemble

I.2.9.1 Utiliser un type Ensemble pour représenter un ensemble de variables dont le nombre est connu et réduit et dont l'ordre n'est pas significatif. Identifier chaque variable par un étiquetage spécifique au contexte.

*Exemple:*

```
Nom-Utilisateur ::= SET
{ nom-personne            [0] IMPLICIT Chaîne-Visible
  nom-Organisation       [1] IMPLICIT Chaîne-Visible
  nom-pays                [2] IMPLICIT Chaîne-Visible }
```

I.2.9.2 Utiliser un type Ensemble avec "OPTIONAL" pour représenter un ensemble de variables qui est un sous-ensemble (au sens strict ou non) d'un autre ensemble de variables dont le nombre est connu et raisonnablement petit, et dont l'ordre n'est pas significatif. Identifier chaque variable par un étiquetage spécifique au contexte.

*Exemple:*

```
Nom-Utilisateur ::= SET
{ nom-personne            [0] IMPLICIT Chaîne-Visible
  nom-organisation       [1] IMPLICIT Chaîne-Visible OPTIONAL
  -- devient par défaut celui de l'organisation locale --

  nom-Pays                [2] IMPLICIT Chaîne-Visible OPTIONAL
  -- devient par défaut celui du pays local -- }
```

I.2.9.3 Utiliser un type Ensemble pour représenter un ensemble de variables dont la composition risque de changer d'une version du protocole à la suivante. Identifier chaque variable par un étiquetage spécifique au contexte.

*Exemple:*

```
Nom-Utilisateur ::= SET
{ nom-Personne            [0] IMPLICIT Chaîne-Visible
  nom-Organisation       [1] IMPLICIT Chaîne-Visible OPTIONAL
  -- devient par défaut celui de l'organisation locale --

  nom-Pays                [2] IMPLICIT Chaîne-Visible OPTIONAL
  -- devient par défaut celui du pays local
  -- les autres attributs facultatifs sont réservés pour étude ultérieure -- }
```

I.2.9.4 Si le nombre de membres d'un ensemble est fixe, affecter un nom de dénotation à chaque membre dont l'usage n'est pas parfaitement évident d'après son type.

*Exemple:*

```
Attributs-Fichier ::= SET
{ titulaire [0] IMPLICIT Nom-Utilisateur
  taille-Contenu-En-Octets [1] IMPLICIT INTEGER
  ... } [2] IMPLICIT Contrôles-Accès
```

I.2.9.5 Utiliser un type Ensemble pour représenter un ensemble de variables dont les types sont identiques et dont l'ordre n'est pas significatif.

*Exemple:*

```
Mots-clé ::= SET OF Chaîne-Visible -- dans un ordre quelconque
```

Voir également les § I.2.5.4 et I.2.10.3.

## I.2.10 *Étiqueté*

I.2.10.1 Utiliser un type Étiqueté universel pour définir un type de données d'usage général, indépendant de l'application, et qui doit être distinguable (par sa représentation) des autres types de données.

*Exemple:*

```
Clé-Chiffrement ::= [UNIVERSAL 30] IMPLICIT OCTET STRING
-- sept octets
```

I.2.10.2 Utiliser un type Étiqueté particulier à l'application pour définir un type de données très répandu et largement utilisé, dans un contexte de présentation particulier, où il doit être distinguable (par sa présentation) de tous les autres types de données qui y sont utilisés.

*Exemple:*

```
Nom-Fichier ::= [APPLICATION 8] IMPLICIT SEQUENCE
{ nom-Répertoire Chaîne-Visible
  nom-Fichier-Relatif-Répertoire Chaîne-Visible }
```

I.2.10.3 Utiliser des types Étiquetés spécifiques au contexte pour distinguer les membres d'un ensemble. Affecter des étiquettes numériques commençant à zéro lorsque la seule contrainte qui leur est imposée est qu'elles soient distinctes.

*Exemple:*

```
Enregistrement-Client ::= SET
{ nom [0] IMPLICIT Chaîne-Visible
  adresse-Postale [1] IMPLICIT Chaîne-Visible
  numéro-Compte [2] IMPLICIT INTEGER
  solde-Dû [3] IMPLICIT INTEGER -- en centimes -- }
```

I.2.10.4 Lorsqu'un membre particulier d'un ensemble a été étiqueté d'une façon particulière à une application, ne pas utiliser d'autres étiquettes spécifiques au contexte, sauf si elle est nécessaire (ou si elle risque de l'être ultérieurement) pour distinguer les membres entre eux. Lorsque le membre d'un ensemble a déjà une étiquette de classe universelle, toute autre étiquette doit être spécifique au contexte.

*Exemple:*

```
Enregistrement-Produit ::= SET
{ description Code-Uniforme [0] IMPLICIT Chaîne-Visible
  numéro-Stock [1] IMPLICIT INTEGER
  niveau-Stock [2] IMPLICIT INTEGER }

Code-Uniforme ::= [APPLICATION 13] IMPLICIT INTEGER
```

I.2.10.5 Utiliser des types Étiquetés spécifiques à un contexte, pour distinguer les possibilités offertes par un CHOICE. Affecter les étiquettes numériques à partir de zéro, si la seule contrainte est qu'elles soient distinctes.

*Exemple:*

```
Attribut-Client ::= CHOICE
{ nom [0] IMPLICIT Chaîne-Visible
  adresse-Postale [1] IMPLICIT Chaîne-Visible
  numéro-Compte [2] IMPLICIT INTEGER
  solde-Dû [3] IMPLICIT INTEGER -- en centimes --}
```

I.2.10.6 Lorsqu'une possibilité particulière d'un CHOICE a été définie en utilisant un type Étiqueté particulier à une application, ne pas utiliser d'autres étiquettes spécifiques au contexte, sauf si elle est nécessaire (ou risque de l'être ultérieurement) pour distinguer les étiquettes.

*Exemple:*

```
Descriptif-Produit ::= CHOICE
{ description Code-Uniforme [0] IMPLICIT Chaîne-Visible
  numéro-Inventaire [1] IMPLICIT INTEGER }
Code-Uniforme ::= [APPLICATION 13] IMPLICIT INTEGER
```

I.2.10.7 Lorsqu'une possibilité particulière offerte par CHOICE a été étiquetée dans la classe universelle, une autre étiquette spécifique au contexte devrait être utilisée, sauf si le choix a pour objet de permettre plusieurs types universels.

*Exemple:*

```
Identificateur-Client ::= CHOICE
{ nom Chaîne-Visible
  numéro INTEGER }
```

I.2.10.8 Utiliser un type Étiqueté à usage privé pour définir un type de données qui trouve son utilisation dans un organisme particulier ou dans un pays déterminé et qui doit pouvoir être distingué (par sa représentation) de tous les autres types de données utilisés par cet organisme ou pays.

*Exemple:*

```
Numéro-Badge-Acme ::= [PRIVATE 2] IMPLICIT INTEGER
```

I.2.10.9 Ces directives utilisent, dans les exemples, l'étiquetage implicite chaque fois qu'il est autorisé. Ceci peut, selon les règles de codage, se traduire par une représentation compacte, ce qui est très souhaitable dans certaines applications. Dans d'autres applications, la compacité peut être moins importante que par exemple la possibilité de procéder à un contrôle poussé du type. Dans ce dernier cas, l'étiquetage explicite peut être utilisé.

Voir également les § I.2.9.1, I.2.9.2, I.2.11.1 et I.2.11.2.

## I.2.11 Choix

I.2.11.1 Utiliser un CHOICE pour représenter une variable qui est choisie dans un ensemble de variables de nombre connu et petit. Identifier chaque variable possible par un étiquetage spécifique au contexte.

*Exemple:*

```
Identificateur-Fichier ::= CHOICE
{ nom-Relatif [0] IMPLICIT Chaîne-Visible
  -- nom de fichier (par exemple, "Rapport-Avancement-Mars")

  nom-Absolu [1] IMPLICIT Chaîne-Visible
  -- noms du fichier et du répertoire qui le contient
  -- (par exemple, "Rapport-Avancement-Mars < Williams > ")

  numéro-Série [2] IMPLICIT INTEGER
  -- identificateur du fichier affecté par le système --}
```

I.2.11.2 Utiliser un CHOICE pour représenter une variable qui est choisie à partir d'un ensemble de variables dont la composition risque de changer d'une version du protocole à la suivante. Identifier chaque variable possible par un étiquetage spécifique au contexte.

*Exemple:*

```
Identificateur-Fichier ::= CHOICE
  {nom-Relatif           [0] IMPLICIT Chaîne-Visible
   -- nom de fichier (par exemple, "Rapport-Avancement-Mars")

  nom-Absolu             [1] IMPLICIT Chaîne-Visible
   -- noms du fichier et du répertoire qui le contient
   -- (par exemple, "Rapport-Avancement-Mars < Williams > ")
   -- identificateur du fichier affecté par le système --}
```

I.2.11.3 Un nom de référence devrait être affecté à chaque possibilité dont l'objet n'est pas parfaitement évident d'après son type.

*Exemple:*

```
Identificateur-Fichier ::= CHOICE
  {nom-Relatif           [0] IMPLICIT Chaîne-Visible
   -- nom de fichier (par exemple, "Rapport-Avancement-Mars")

  nom-Absolu             [1] IMPLICIT Chaîne-Visible
   -- noms du fichier et du répertoire qui le contient
   -- (par exemple, "Rapport-Avancement-Mars < Williams > ")

                                [2] IMPLICIT Numéro-Série
   -- identificateur du fichier affecté par le système --}
```

I.2.11.4 Lorsque l'étiquetage implicite est la règle dans une application particulière de la présente Recommandation, utiliser un CHOICE d'un seul type, lorsqu'il est envisagé d'avoir ultérieurement plusieurs types. Le risque de devoir recourir à un étiquetage implicite est ainsi écarté, ce qui facilite la transition.

*Exemple:*

```
Voeux ::= [APPLICATION 12] CHOICE
          {Chaîne-Visible}
```

en prévision de:

```
Voeux ::= [APPLICATION 12] CHOICE
          {Chaîne-Visible
           Voix}
```

## I.2.12 Type Sélection

I.2.12.1 Utiliser un type Sélection pour représenter une variable dont le type est celui de certaines possibilités particulières offertes par un CHOICE défini antérieurement.

I.2.12.2 Considérons la définition:

```
Attribut-Fichier ::= CHOICE
  {date-dernière-utilisation INTEGER
   nom-fichier           Chaîne-Visible}
```

La définition suivante est possible:

```
Attributs-Courants ::= SEQUENCE
  {date-dernière-utilisation < Attribut-Fichier
   nom-fichier              < Attribut-Fichier}
```

avec la notation de valeur possible:

```
{date-dernière-utilisation 27
 nom-fichier "PROGRAM"}
```

La définition suivante est également possible:

```
Liste-Attributs ::= SEQUENCE
{ premier-attribut date-dernière-utilisation < Attribut-Fichier
  second-attribut nom-fichier < Attribut-Fichier }
```

avec la notation de valeur possible:

```
{ premier-attribut 27,
  second-attribut "PROGRAM" }
```

### I.2.13 *Quelconque*

I.2.13.1 Utiliser un type Quelconque pour représenter une variable dont le type n'est pas spécifié, ou spécifié ailleurs en ASN.1.

*Exemple:*

```
Contenu-Messages ::= ANY
-- élément de données dont le type n'est pas spécifié
-- en dehors de cette norme en notation ASN.1
```

### I.2.14 *Externe*

I.2.14.1 Utiliser un type Externe pour représenter une variable dont le type n'est pas spécifié, ou spécifié ailleurs sans restriction quant à la notation utilisée pour le spécifier.

*Exemple:*

```
Contenu-Fichiers ::= EXTERNAL
Liste-Documents ::= SEQUENCE OF EXTERNAL
```

### I.3 *Exemple d'utilisation de la notation de macro*

Supposons que l'on désire avoir une notation de définition de type de la forme:

```
PAIR TYPEX = .... TYPEY = ....
```

avec une notation de valeur correspondante autorisant:

```
(X = ----, Y = ----)
```

Les .... et les ---- dénotent respectivement tout type ASN.1 et la valeur correspondante.

Cette notation de type de macro pourrait être utilisée pour définir des types et des valeurs comme suit:

```
T1 ::= PAIR
  TYPEX = INTEGER
  TYPEY = BOOLEAN
```

```
T2 ::= PAIR
  TYPEX = Chaîne-Visible
  TYPEY = T1
```

Une valeur du type T1 pourrait alors être:

```
(X = 3, Y = TRUE)
```

et une valeur du type T2 pourrait être:

```
(X = "Nom", Y = (X = 4, Y = FALSE))
```

La définition de macro suivante pourrait être utilisée pour établir cette nouvelle notation, comme extension de la notation ASN.1 de base:

```
PAIR
MACRO ::= BEGIN
TYPE NOTATION ::=
    "TYPEX"
    "=="
    type (Type-local-1)
    -- attend n'importe quel type ASN.1 et l'affecte
    -- à la variable Type-local-1
    "TYPEY"
    "=="
    type (Type-local-2)
    -- attend un second type ASN.1 et l'affecte
    -- à la variable Type-local-2

VALUE NOTATION ::=
    "("
    "X"
    "=="
    value (Valeur-locale-1 Type-local-1)
        -- attend une valeur pour le type de
        -- Type-local-1 et l'affecte
        -- à la variable Valeur-locale-1
    ","
    "Y"
    "=="
    value (Valeur-locale-2 Type-local-2)
        -- attend une valeur pour le type de
        -- Type-local-2 et l'affecte
        -- à la variable Valeur-locale-2
    < VALUE SEQUENCE {Type-local-1, Type-local-2}
        ::= {Valeur-locale-1, Valeur-locale-2}
    -- cette "définition incluse" renvoie
    -- la valeur finale comme valeur
    -- d'une séquence des deux types
    ")"

END
```

Dans cet exemple, le type ASN.1 de base de la valeur renvoyée est indépendant de l'occurrence effective de la notation de valeur, mais dépend de l'occurrence de la notation de type qui est utilisée. Dans d'autres cas, il peut être entièrement déterminé par la notation de macro, ou peut dépendre de l'occurrence de la notation de valeur qui est utilisée. A noter, toutefois, que dans tous les cas, c'est la production "VALUE NOTATION" qui doit être examinée pour déterminer le type de la valeur renvoyée. La production "TYPE NOTATION" définit simplement la syntaxe de la définition du type et détermine la valeur initiale des variables locales à utiliser, lors de l'analyse d'une occurrence de la notation de valeur.

#### I.4 Utilisation d'ASN.1 pour l'identification des syntaxes abstraites

- I.4.1 L'utilisation du service de présentation (Recommandation X.216) requiert la spécification des valeurs appelées *valeurs de données de présentation*, et le regroupement de celles-ci en ensembles appelés *syntaxes abstraites*. Chacun de ces ensembles reçoit un *nom de syntaxe abstraite* de l'identificateur d'objet type ASN.1.
- I.4.2 L'ASN.1 peut servir d'instrument général pour la spécification des valeurs de données de présentation et leur regroupement en syntaxes abstraites nommées.
- I.4.3 Dans l'utilisation la plus simple, il n'y a qu'un seul type d'ASN.1, de telle sorte que chaque valeur de données de présentation de la syntaxe abstraite nommée est une valeur de ce type ASN.1. Ce type doit normalement être un type choix, et chaque valeur de données de présentation doit être un type possible de ce type choix. Dans ce cas, il est recommandé que la notation de module ASN.1 soit utilisée pour contenir ce type choix comme premier

type défini, suivi par la définition des types (non universels) dénotés directement ou indirectement par ce type choix.

*Remarque* – Cette méthode n'est pas destinée à exclure les dénnotations de types définis dans d'autres modules.

I.4.4 On trouvera ci-dessous un exemple de texte qui pourrait figurer dans une norme d'application. L'exemple se conclut par la mention "END OF EXAMPLE", afin d'éviter toute confusion.

*Exemple:*

```
ISOxxxx-yyyy DEFINITIONS ::=
BEGIN
PDU ::= CHOICE
    { connexion-pdu     ...,
      données-pdu CHOICE
        {
          ... }
        ,
      ...
    }
...
END
```

Cette norme internationale affecte la valeur d'identificateur d'objet ASN.1

{Norme ISO xxxx syntaxe-abstraite (1)}

comme nom de syntaxe abstraite à l'ensemble de valeurs de données de présentation dont chacune est une valeur du type ASN.1 "ISOxxxx-yyyy . PDU".

La valeur du descripteur d'objet ASN.1 correspondante sera

"....."

Les valeurs ASN.1 d'identificateur d'objet et de descripteur d'objet

{joint-iso-ccitt asn1 (1) codage-de-base (1)}

et

"Codage de base d'un type ASN.1 unique"

(affectées à un objet informationnel dans la Recommandation X.209) peuvent servir de nom de syntaxe de transfert avec ce nom de syntaxe abstraite.

END OF EXAMPLE

I.4.5 La norme peut en outre servir de support à la syntaxe de transfert obtenue en imposant

{joint-iso-ccitt asn1 (1) codage-de-base (1)}

à cette syntaxe abstraite.

I.5 *Sous-types*

I.5.1 Utiliser des sous-types pour limiter les valeurs d'un type existant autorisées dans un cas particulier.

*Exemple:*

Nombre-Atomique ::= INTEGER (1..104)

Chaîne-Tonalités-Clavier ::= Chaîne IA5 (FROM  
("0" | "1" | "2" | "3" | "4" | "5" | "6" |  
"7" | "8" | "9" | "\*" | "#") |  
SIZE (1..63))

Liste-Paramètres ::= SET SIZE (0..63) OF paramètre

Petits-Nombres-Premiers ::= INTEGER (2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29)

I.5.2 Lorsque deux types apparentés ou plus ont de nombreux points en commun, il faut envisager de définir explicitement leur parent commun comme un type, et de se servir de sous-types pour chacun des types. Cette solution

permet de rendre clairs leurs relations et leurs points communs, et les favorise (sans contrainte) au fur et à mesure que les types évoluent. Elle facilite donc des méthodes communes de mise en oeuvre pour le traitement des valeurs de ces types.

*Exemple:*

```
Enveloppe ::= SET {typeA TypeA,  
                  typeB TypeB OPTIONAL,  
                  typeC TypeC OPTIONAL}  
-- le parent commun
```

```
Enveloppe-AB ::= Enveloppe (WITH COMPONENTS  
                           { ...,  
                           typeB PRESENT, typeC ABSENT})  
-- où le typeB doit toujours  
-- apparaître mais pas le typeC
```

```
Enveloppe-AC ::= Enveloppe (WITH COMPONENTS  
                           { ...,  
                           typeB ABSENT, typeC PRESENT})  
-- où le typeC doit toujours  
-- apparaître mais pas le typeB
```

Ces dernières définitions pourraient aussi être exprimées comme suit:

```
Enveloppe-AB ::= Enveloppe (WITH COMPONENTS {typeA,typeB})
```

```
Enveloppe-AC ::= Enveloppe (WITH COMPONENTS {typeA,typeC})
```

Le choix entre les différentes solutions s'appuierait sur des facteurs comme le nombre de composants dans le type parent et le nombre de ceux qui sont optionnels, l'ampleur des différences entre les différents types et la stratégie probable de l'évolution.

I.5.3 Utiliser des sous-types pour définir partiellement une valeur, par exemple une unité de données de protocole qui doit être soumise à un essai de conformité, ne portant que sur certains éléments du PDU.

*Exemple:*

Etant donné:

```
PDU ::= SET  
  
        { alpha    [0]    INTEGER  
          bêta     [1]    Chaîne-IA5 OPTIONAL  
          gamma    [2]    SEQUENCE OF Paramètre  
          delta    [3]    BOOLEAN}
```

puis en composant un essai qui nécessite que le Booléen soit faux et l'Entier négatif, écrire:

```
Essai-PDU ::= PDU (WITH COMPONENTS  
                  { ...,  
                  delta (FALSE)  
                  alpha (MIN...<0)})
```

et si, par ailleurs, bêta-la-Chaîne-IA5 doit être présente et longue de 5 ou 12 caractères, écrire:

```
Essai-PDU-Supplémentaire ::= Essai-PDU (WITH COMPONENTS  
                                   { ...,  
                                   bêta (SIZE (5/12)) PRESENT})
```

I.5.4 Si un type de données d'usage général a été défini comme une SEQUENCE OF, utiliser des sous-types pour définir un sous-type restreint du type général:

*Exemple:*

```
Bloc-texte ::= SEQUENCE OF Chaîne-Visible
Adresse ::= Bloc-texte
              (SIZE (1..6) |
              WITH COMPONENT (SIZE(1..32)))
```

I.5.5 Utiliser des sous-types contenus pour former de nouveaux sous-types à partir des sous-types existants:

*Exemple:*

```
Mois ::= ENUMERATED {
    janvier (1),
    février (2),
    mars (3),
    avril (4),
    mai (5),
    juin (6),
    juillet (7),
    août (8),
    septembre (9),
    octobre (10),
    novembre (11),
    décembre (12)}

Premier-Trimestre ::= Mois (
    janvier |
    février |
    mars)

Deuxième-Trimestre ::= Mois (
    avril |
    mai |
    juin)

Troisième-Trimestre ::= Mois (
    juillet |
    août |
    septembre)

Quatrième-Trimestre ::= Mois (
    octobre |
    novembre |
    décembre)

Premier-Semestre ::= Mois (
    INCLUDES Premier-trimestre |
    INCLUDES Deuxième-trimestre)

Deuxième-Semestre ::= Mois (
    INCLUDES Troisième-trimestre |
    INCLUDES Quatrième-trimestre)
```

## APPENDICE II

(à la Recommandation X.208)

### Résumé de la notation ASN.1

Les items ci-dessous sont définis dans le § 8:

dénotation-type	INTEGER	BEGIN
identificateur	BIT	END
dénotation-valeur	STRING	DEFINITIONS
dénotation-module	OCTET	EXPLICIT
commentaire	NULL	ENUMERATED
vide	SEQUENCE	EXPORTS
nombre	OF	IMPORTS
chaîne-b	SET	
chaîne-h	IMPLICIT	REAL
chaîne-c	CHOICE	INCLUDES
"::="	ANY	MIN
{	EXTERNAL	MAX
}	OBJECT	SIZE
<	IDENTIFIER	FROM
,	OPTIONAL	WITH
.	DEFAULT	COMPONENT
(	COMPONENTS	PRESENT
)	UNIVERSAL	ABSENT
[	APPLICATION	DEFINED
]	PRIVATE	BY
-	TRUE	PLUS-INFINITY
BOOLEAN	FALSE	MINUS-INFINITY

Les productions ci-dessous sont employées dans la présente Recommandation, avec les items ci-dessus comme symboles terminaux:

Définition-Module ::=  
 Identificateur-Module  
 DEFINITIONS  
 Défaut-Etiquette  
 "::="

BEGIN

Corps-Module

END

Défaut-Etiquette ::=  
 EXPLICIT TAGS |  
 IMPLICIT TAGS |  
 vide

Identificateur-Module ::=  
 Dénotation-Module  
 Identificateur-Affecté

Identificateur-Affecté ::=  
 Valeur-Identificateur-Objet |  
 vide

Corps-Module ::=  
 Exportations Importations Liste-Affectations |  
 vide

Exportations ::=  
 EXPORTS Symboles-Exportés; |  
 vide

Symboles-Exportés ::=  
     Liste-Symboles |  
     vide

Importations ::=  
     IMPORTS Symboles-Importés; |  
     vide

Symboles-Importés ::=  
     Symboles-De-Liste-Modules |  
     vide

Symboles-De-Liste-Modules ::=  
     Symboles-De-Module Symboles-De-Liste-Modules |  
     Symboles-De-Module

Symboles-De-Module ::=  
     Liste-Symboles FROM Identificateur-Module

Liste-Symboles ::= Symbole, Liste-Symboles | Symbole

Symbole ::= Dénotation-Type | Dénotation-Valeur

Liste-Affectations ::=  
     Affectation Liste-Affectations |  
     Affectation

Affectation ::= Affectation-Type | Affectation-Valeur

Dénotation-Type-Externe ::=  
     Dénotation-Module  
     .  
     Dénotation-Type

Dénotation-Valeur-Externe ::=  
     Dénotation-Module  
     .  
     Dénotation-Valeur

Type-Défini ::= Dénotation-Type-externe | Dénotation-Type

Valeur-Définie ::= Dénotation-Valeur-Externe | Dénotation-Valeur

Affectation-Type ::=  
     Dénotation-Type  
     "::="

    Type

Affectation-Valeur ::=  
     Dénotation-Valeur  
     Type  
     "::="

    Valeur

Type ::= Type-Prédéfini | Type-Défini | Sous-Type

Type-Prédéfini ::=

Type-Booléen  
Type-Entier  
Type-Chaîne-Binaire  
Type-Chaîne-Octets  
Type-Vide  
Type-Séquence  
Type-Séquence-De  
Type-Ensemble  
Type-Ensemble-De  
Type-Choix  
Type-Sélection  
Type-Etiqueté  
Type-Quelconque  
Type-Identificateur-Objet  
Type-Chaîne-Caractères  
Type-Utile  
Type-Énuméré  
Type-Réel

Type-Nommé ::= Type-identificateur | Type | Type-Sélection

Valeur ::= Valeur-Prédéfini | Valeur-Défini

Valeur-Prédéfini ::=

Valeur-Booléen  
Valeur-Entier  
Valeur-Chaîne-Binaire  
Valeur-Chaîne-Octets  
Valeur-Vide  
Valeur-Séquence  
Valeur-Séquence-De  
Valeur-Ensemble  
Valeur-Ensemble-De  
Valeur-Choix  
Valeur-Sélection  
Valeur-Etiqueté  
Valeur-Quelconque  
Valeur-Identificateur-Objet  
Valeur-Chaîne-Caractères  
Valeur-Énuméré  
Valeur-Réel

Valeur-Nommé ::= Valeur-identificateur | Valeur

Type-Booléen ::= BOOLEAN

Valeur-Booléen ::= TRUE | FALSE

Type-Entier ::= INTEGER | INTEGER {Liste-Nombres-Nommés}

Liste-Nombres-Nommés ::=

Nombres-Nommés |  
Liste-Nombres-Nommés, Nombres-Nommés

Nombres-Nommés ::=

Identificateur (Nombre-Signé) |  
Identificateur (Valeur-Définie)

Nombre-Signé ::= Nombre | -Nombre

Valeur-Entier ::= Nombre-Signé | Identificateur  
 Type-Énuméré ::= ENUMERATED {Énumération}  
 Énumération ::=  
     Nombre-Nommé |  
     Nombre-Nommé, Énumération  
 Valeur-Énuméré ::= Identificateur  
 Type-Réel ::= REAL  
 Valeur-Réel ::= Valeur-Réel-Numérique | Valeur-Réel-Spécial  
 Valeur-Réel-Numérique ::= {Mantisse, Base, Exposant} | 0  
 Mantisse ::= Nombre-Signé  
 Base ::= 2 | 10  
 Exposant ::= Nombre-Signé  
 Valeur-Réel-Spécial ::= PLUS-INFINITY | MINUS-INFINITY  
 Type-Chaîne-Binaire ::= BIT STRING | BIT STRING {Liste-Bits-Nommés}  
 Liste-Bits-Nommés ::= Bit-Nommé | Liste-Bits-Nommés, Bit-Nommé  
 Bit-Nommé ::=  
     Identificateur (Nombre) |  
     Identificateur (valeur-Définie)  
 Valeur-Chaîne-Binaire ::= Chaîne-b | Chaîne-h | {Liste-Identificateurs} | {}  
 Liste-Identificateurs ::= Identificateur | Liste-Identificateurs, Identificateur  
 Type-Chaîne-Octets ::= OCTET STRING  
 Valeur-Chaîne-Octets ::= Chaîne-b | Chaîne-h  
 Type-Vide ::= NULL  
 Valeur-Vide ::= NULL  
 Type-Séquence ::=  
     SEQUENCE {Liste-Types-Éléments} |  
     SEQUENCE {}  
 Liste-Types-Éléments ::=  
     Type-Élément |  
     Liste-Types-Éléments, Type-Élément  
 Type-Élément ::=  
     Type-Nommé |  
     Type-Nommé OPTIONAL |  
     Type-Nommé DEFAULT Valeur |  
     COMPONENTS OF Type  
 Valeur-Séquence ::= {Liste-Valeurs-Éléments} | {}  
 Liste-Valeurs-Éléments ::=  
     Valeur-Nommée |  
     Liste-Valeurs-Éléments, Valeur-Nommée  
 Type-Séquence-De ::= SEQUENCE OF Type | SEQUENCE  
 Valeur-Séquence-De ::= {Liste-Valeurs} | {}  
 Liste-Valeurs ::= Valeur | Liste-Valeurs, Valeur  
 Type-Ensemble ::= SET {Liste-Types-Éléments} | SET {}  
 Valeur-Ensemble ::= {Liste-Valeurs-Éléments} | {}  
 Type-Ensemble-De ::= SET OF Type | SET

Valeur-Ensemble-De ::= {Liste-Valeurs} | {}  
 Type-Choix ::= CHOICE {Liste-Types-Possibles}  
 Liste-Types-Possibles ::=  
     Type-Nommé |  
     Liste-Types-Possibles, Type-Nommé  
 Valeur-Choix ::= Valeur-Nommée  
 Type-Sélection ::= Identificateur < Type  
 Valeur-Sélection ::= Valeur-Nommée  
 Type-Etiqueté ::=  
     Type-Etiquette |  
     Type-Etiquette IMPLICIT |  
     Type-Etiquette EXPLICIT  
 Etiquette ::= [Classe Numéro-Classe]  
 Numéro-Classe ::= Numéro | Valeur-Définie  
 Classe ::=  
     UNIVERSAL |  
     APPLICATION |  
     PRIVATE |  
     vide  
 Valeur-Etiqueté ::= Valeur  
 Type-Quelconque ::=  
     ANY |  
     ANY DEFINED BY Identificateur  
 Valeur-Quelconque ::= Valeur-Type  
 Type-Identificateur-Objet ::= OBJECT IDENTIFIER  
 Valeur-Identificateur-Objet ::=  
     {Liste-Composants-id-Ob} |  
     {Valeur-Définie Liste-Composants-id-Ob}  
 Liste-Composants-id-Ob ::=  
     Composant-id-Ob |  
     Composant-id-Ob Liste-Composants-id-Ob  
 Composant-id-Ob ::=  
     Forme-Nom |  
     Forme-Numéro |  
     Forme-Nom-Et-Numéro  
 Forme-Nom ::= Identificateur  
 Forme-Numéro ::= Numéro | Valeur-Défini  
 Forme-Nom-Et-Numéro ::= Identificateur (Forme-Numéro)  
 Type-Chaîne-Caractères ::= Dénotation-Type  
 Valeur-Chaîne-Caractères ::= Chaîne-c  
 Type-Utile ::= Dénotation-Type

Les types de chaîne de caractères ci-dessous sont définis dans la section 2:

Chaîne-Numérique	Chaîne-Visible
Chaîne-Imprimable	Chaîne-ISO 646
Chaîne-Télex	Chaîne-IA5
Chaîne-T61	Chaîne-Graphique
Chaîne-Vidéotex	Chaîne-Générale

Les types utiles ci-dessous sont définis dans la section 3:

Heure-Généralisée                      EXTERNAL  
Heure-UTC                                Descripteur-Objet

Les productions ci-dessous sont employées dans la section 4:

Sous-Type ::=

Type-Parent Spécif-Sous-type |  
SET Limitation-Taille OF Type |  
SEQUENCE Limitation-Taille OF Type

Type-Parent ::= Type

Spécif-Sous-type ::=

(Ensemble-Valeurs-Sous-type Liste-Ensemble-Valeurs-Sous-type)

Liste-Ensemble-Valeurs-Sous-type ::=

" | "  
Ensemble-Valeurs-Sous-type  
Liste-Ensemble-Valeurs-Sous-type |  
vide

Ensemble-Valeurs-Sous-type ::=

Valeur-Unique |  
Sous-Type-Contenu |  
Plage-Valeur |  
Alphabet-Autorisé | Limitation-Taille | Contraintes-Types-Internes  
Valeur-unique ::= Valeur  
Sous-Type-Contenu ::= INCLUDES Type  
Plage-Valeur ::= Limite-Inférieure .. Limite-Supérieure

Limite-inférieure ::= Limite-Inférieure | Limite-Inférieure <

Limite supérieure ::= Limite-Supérieure | < Limite-Supérieure

Limite-Inférieure ::= Valeur | MIN

Limite-Supérieure ::= Valeur | MAX

Limitation-Taille ::= SIZE Spécif-Sous-type

Alphabet-Autorisé ::= FROM Spécif-Sous-type

Contraintes-Type-Interne ::=

WITH COMPONENT Contrainte-Type-Unique |

WITH COMPONENTS Contraintes-Types-Multiples

Contrainte-Type-Unique ::= Spécif-Sous-type

Contraintes-Types-Multiples ::=

Spécification-Complète | Spécification-Partielle

Spécification-Complète ::= { Contraintes-Types }

Spécification-Partielle ::= { ..., Contraintes-Types }

Contraintes-Types ::=

Contrainte-Nommée |  
Contrainte-Nommée, Contraintes-Types

Contrainte-Nommée ::= Identificateur Contrainte | Contrainte

Contrainte ::= Contrainte-Valeur Contrainte-Présence

Contrainte-Valeur ::= Spécif-Sous-type | vide

Contrainte-Présence ::= PRESENT | ABSENT | vide | OPTIONAL

Les items supplémentaires ci-dessous sont définis au § A.2 en vue de leur utilisation dans la notation de macro:

dénotation-macro	"nombre"
dénotation-production	"vide"
dénotation-type-local	MACRO
dénotation-valeur-locale	TYPE
"   "	NOTATION
>	VALUE
chaîne-a	valeur
"chaîne"	type
"identificateur"	

Les productions ci-dessous sont utilisées dans l'annexe A, avec les items ci-dessus et les items énumérés au début du présent appendice, comme symboles terminaux:

```

Définition-Macro ::=
    dénotation-macro
    MACRO
    "::~="
    Substance-Macro

Substance-Macro ::=
    BEGIN Corps-Macro END |
    dénotation-macro |
    Dénotation-macro-externe

Corps-Macro ::=
    Production-Type
    Production-Valeur
    Productions-Support

Production-Type ::=
    TYPE NOTATION
    "::~="
    Liste-Macros-Possibles

Production-Valeur ::=
    VALUE NOTATION
    "::~="
    Liste-Macros-Possibles

Productions-Support ::= Liste-Productions | vide
Liste-Productions ::= Production | Liste-Productions Production
Production ::=
    dénotation-production
    "::~="
    Liste-Macros-Possibles

Dénotation-macro-externe ::=
    dénotation-module . dénotation-macro

Liste-Macros-Possibles ::=
    Macro-Possible |
    Liste-Macros-Possibles " | " Macro-Possible

Macro-Possible ::= Liste-Symboles
Liste-Symboles ::= Elément-Symbole | Liste-Symboles Elément-Symbole
Elément-Symbole ::= Déf-Symbole | Définitions-incluses

```

Déf-Symbole ::=  
     chaîne-a |  
     dénotation-production |  
     "chaîne" |  
     "identificateur" |  
     "nombre" |  
     "vide" |  
     type |  
     type(dénotation-type-local) |  
     valeur(Type-Macro) |  
     valeur(dénotation-valeur-locale Type-Macro) |  
     valeur(VALUE Type-Macro)

Type-Macro ::= dénotation-type-local | Type

Définitions-includes ::= < Liste-Définitions-includes >

Liste-Définitions-includes ::=  
     Définition-include |  
     Liste-Définitions-includes Définition-include

Définition-include ::=  
     Affectation-Type-Local |  
     Affectation-Valeur-Locale

Affectation-Type-Local ::=  
     dénotation-type-local  
     "::="

    Type-Macro

Affectation-Valeur-Locale ::=  
     dénotation-valeur-locale  
     Type-Macro  
     "::="

    Valeur-Macro

Valeur-Macro ::= Valeur | dénotation-valeur-locale