



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.681

(12/97)

SÉRIE X: RÉSEAUX POUR DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS

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

**Technologies de l'information – Notation de
syntaxe abstraite numéro un: spécification
des objets informationnels**

Recommandation UIT-T X.681

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE X

RÉSEAUX POUR DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

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

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

NORME INTERNATIONALE 8824-2

RECOMMANDATION UIT-T X.681

**TECHNOLOGIES DE L'INFORMATION –
NOTATION DE SYNTAXE ABSTRAITE NUMÉRO UN:
SPÉCIFICATION DES OBJETS INFORMATIONNELS**

Résumé

La présente Recommandation | Norme internationale spécifie la notation ASN.1 qui permet de définir les classes d'objets informationnels ainsi que les objets informationnels proprement dits et de leur donner des noms de référence. Une classe d'objets informationnels est un modèle utilisé pour représenter un ensemble d'informations, qui définit les propriétés de tous les membres de cette classe.

Source

La Recommandation X.681 de l'UIT-T a été approuvée le 12 décembre 1997. Un texte identique est publié comme Norme internationale ISO/CEI 8824-2.

AVANT-PROPOS

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

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

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

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

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

© UIT 1998

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

TABLE DES MATIÈRES

		<i>Page</i>
1	Domaine d'application	1
2	Références normatives	1
	2.1 Recommandations Normes internationales identiques	1
3	Définitions.....	1
	3.1 Spécification de la notation de base	1
	3.2 Spécification des contraintes.....	1
	3.3 Paramétrage des spécifications ASN.1	2
	3.4 Autres définitions.....	2
4	Abréviations	3
5	Conventions	3
6	Notation.....	3
	6.1 Affectations.....	3
	6.2 Types	4
	6.3 Valeurs.....	4
	6.4 Eléments	4
7	Unités lexicales ASN.1	4
	7.1 Référence de classe d'objets informationnels	4
	7.2 Référence d'objets informationnels	4
	7.3 Référence d'ensemble d'objets	5
	7.4 Référence de champ de type	5
	7.5 Référence de champ de valeur	5
	7.6 Référence de champ d'ensemble de valeurs.....	5
	7.7 Référence de champ d'objet.....	5
	7.8 Référence de champ d'ensemble d'objets.....	5
	7.9 Mot	5
	7.10 Autres unités lexicales mots clés.....	5
8	Définitions de références.....	6
9	Définition et affectation d'une classe d'objets informationnels	6
	9.16 Exemples.....	10
10	Syntaxe conviviale	10
	10.13 Exemples.....	12
11	Définition et affectation des objets informationnels.....	13
	11.9 Exemples (syntaxe par défaut)	14
	11.10 Exemples (syntaxe conviviale).....	14
12	Définition et affectation des ensembles d'objets informationnels.....	15
	12.7 Exemple	16
13	Tables associées	16
	13.5 Exemples de noms de champs «FieldName» valides.....	16
14	Notation du type champ de classe d'objets (object class field type)	17
	14.9 Exemple d'utilisation de type champ de classe d'objets «ObjectClassFieldType».....	18
15	Extraction d'information à partir d'objets.....	18
	15.10 Exemple d'information extraite d'objets.....	20

	<i>Page</i>
Annexe A – Classe d’objets informationnels «TYPE-IDENTIFIER» (identificateur de type).....	21
A.4 Exemple	21
Annexe B – Définition d’une syntaxe abstraite	22
B.4 Exemple	22
Annexe C – Le type instance-de (instance-of type)	23
C.9 Exemple	24
Annexe D – Exemples	25
D.1 Exemple d’utilisation de la classe «OPERATION» simplifiée.....	25
D.2 Exemple d’utilisation du type champ de classe d’objets «ObjectClassFieldType».....	26
D.3 Illustration de l’utilisation des objets et des ensembles d’objets.....	27
Annexe E – Annexe didactique sur le modèle ASN.1 d’extension des ensembles d’objets	28
E.1 Ensembles d’objets extensibles.....	28
Annexe F – Récapitulation de la notation.....	29

Introduction

Le concepteur d'applications est fréquemment appelé à concevoir un protocole destiné à fonctionner avec un certain nombre d'instances d'une certaine classe d'objets informationnels, ces instances pouvant être définies par diverses entités au cours du temps. Comme exemple de classe d'objets informationnels, on peut citer les «opérations» du service d'opérations distantes ROS et les «attributs» de l'annuaire de l'OSI.

La présente Recommandation | Norme internationale spécifie une notation permettant de définir des classes d'objets informationnels ainsi que des objets informationnels individuels et des ensembles d'objets informationnels et de leur attribuer des noms de référence.

Une classe d'objets informationnels est caractérisée par les diverses sortes de champs de ses instances. Un champ peut contenir:

- un type quelconque (champ de type);
- une valeur unique d'un type spécifié (champ de valeur de type fixé);
- une valeur unique d'un type spécifié dans un champ de type nommé (champ de valeur de type variable);
- un ensemble non vide de valeurs d'un type spécifié (champ d'ensemble de valeurs de type fixé);
- un ensemble non vide de valeurs d'un type spécifié dans un champ de type nommé (champ d'ensemble de valeurs de type variable);
- un objet informationnel unique pris dans une classe d'objets informationnels spécifiée (champ d'objet);
- un ensemble d'objets informationnels pris dans une classe d'objets informationnels spécifiée (champ d'ensemble d'objets).

On peut choisir un champ de valeur de type fixé d'une classe d'objets informationnels pour identifier sans équivoque les objets informationnels de cette classe. Ce champ est alors appelé le champ identificateur de cette classe. Les valeurs de ce champ, lorsqu'il existe, sont nécessairement uniques pour tout ensemble d'objets informationnels défini dans cette classe. Elles peuvent, mais pas nécessairement, servir à identifier de manière non ambiguë les objets informationnels de cette classe dans un cadre plus large, en adoptant notamment le type identificateur d'objet pour le champ identificateur.

On définit une classe d'objets informationnels en spécifiant:

- le nom des champs;
- la forme de chaque champ (type, valeur de type fixé, valeur de type variable, ensemble de valeurs de type fixé, ensemble de valeurs de type variable, objet ou ensemble d'objets);
- la nature optionnelle ou non des champs et leurs valeurs par défaut;
- le champ identificateur s'il existe.

On définit un objet informationnel individuel d'une classe en fournissant les informations nécessaires à chaque champ.

La notation définie ici permet de spécifier un type ASN.1 en faisant référence à un champ d'une classe d'objets informationnels donnée – le type champ de classe d'objets. La notation donnée dans la Rec. UIT-T X.682 | ISO/CEI 8824-3 permet de restreindre ce type en faisant référence à un ensemble d'objets informationnels spécifique.

Il peut être utile de considérer que la définition d'une classe d'objets informationnels définit la forme d'une table conceptuelle sous-jacente (la table associée) comportant une colonne par champ, chaque ligne de cette table définissant un objet informationnel. La forme de cette table (déterminée par la spécification de la classe d'objets informationnels) détermine la nature des informations à collecter et à utiliser pour compléter une spécification de protocole donnée. Cette table conceptuelle sous-jacente assure la liaison entre ceux qui spécifient les objets informationnels de cette classe et le protocole ayant besoin de ces informations pour que sa spécification soit complète. En général, l'ensemble effectif d'objets informationnels utilisé pour compléter la spécification d'un protocole particulier constitue un paramètre de ce protocole (voir la Rec. UIT-T X.683 | ISO/CEI 8824-4).

La notation «InformationFromObjects» renvoyant à un objet ou un ensemble d'objets donné (probablement un paramètre) peut servir à extraire l'information contenue dans les cellules des tables conceptuelles.

La présente Recommandation | Norme internationale:

- spécifie une notation servant à définir une classe d’objets informationnels et à l’identifier par un nom de référence (voir l’article 9);
- spécifie une notation permettant, lors de la définition d’une classe d’objets informationnels, d’établir une syntaxe conviviale pour définir des objets informationnels de cette classe; une notation par défaut est indiquée pour les classes pour lesquelles aucune syntaxe conviviale n’a été établie (voir l’article 10);
- spécifie une notation servant à définir un objet informationnel et à lui affecter un nom de référence (voir l’article 11), ainsi qu’une notation analogue pour un ensemble d’objets (voir l’article 12);
- définit la «table associée» pour un objet ou d’un ensemble d’objets d’une classe (voir l’article 13);
- spécifie une notation pour le type champ de classe d’objets et ses valeurs (voir l’article 14);
 - NOTE – Ces structures permettent de spécifier un type ASN.1 à l’aide d’un champ nommé ou d’une classe d’objets informationnels nommée. Les contraintes imposées à ce type afin d’en limiter les valeurs à un ensemble d’objets informationnels spécifique figurent dans la Rec. UIT-T X.682 | ISO/CEI 8824-3.
- spécifie une notation pour extraire des informations contenues dans les objets (voir l’article 15).

L’ensemble des objets informationnels utilisés dans la définition d’un ensemble d’objets peut être partiellement ou entièrement inconnu lors de la définition des spécifications ASN.1. Tel est par exemple le cas, dans la gestion d’un réseau, lorsque l’ensemble des objets gérés varie lors de l’exécution des fonctions du gestionnaire de réseau. La présente Recommandation | Norme internationale spécifie les règles d’inclusion d’un **marqueur d’extension** dans la définition de tels ensembles d’objets, pour signaler aux réalisateurs de mises en œuvre que le concepteur s’est abstenu intentionnellement de définir entièrement l’ensemble d’objets dans la spécification ASN.1. Lorsqu’un ensemble d’objets comporte dans sa définition un marqueur d’extension, le réalisateur d’une mise en œuvre doit assurer les moyens – ne relevant pas nécessairement de la notation ASN.1 – permettant d’ajouter dynamiquement des objets à cet ensemble d’objets, et de supprimer des objets qui lui auraient été ajoutés.

L’Annexe A, qui fait partie intégrante de la présente Recommandation | Norme internationale, spécifie la classe d’objets informationnels dont la référence de classe d’objets est «TYPE-IDENTIFIER» (identificateur de type). Il s’agit de la classe utile la plus simple; elle ne comporte que deux champs: un champ identificateur de type identificateur d’objet («OBJECT IDENTIFIER») et un champ de type unique définissant le type ASN.1 servant à acheminer toute l’information concernant un objet particulier de la classe. Cette classe est définie ici en raison de l’utilisation très fréquente des objets informationnels de cette sorte.

L’Annexe B, qui fait partie intégrante de la présente Recommandation | Norme internationale, spécifie la notation servant à définir une syntaxe abstraite (composée de l’ensemble de valeurs d’un même type ASN.1) en définissant un objet informationnel approuvé.

L’Annexe C, qui fait partie intégrante de la présente Recommandation | Norme internationale, spécifie la notation du type instance-de («INSTANCE OF»), qui est capable de véhiculer toute valeur de tout objet informationnel appartenant à une classe spécifiée (qui doit avoir été définie en termes d’identificateur de type «TYPE-IDENTIFIER»).

L’Annexe D, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, fournit quelques exemples sur la manière d’utiliser la notation décrite dans la présente Recommandation | Norme internationale.

L’Annexe E, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, récapitule le modèle ASN.1 d’extension des ensembles d’objets.

L’Annexe F, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, récapitule la notation définie dans la présente Recommandation.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION –
NOTATION DE SYNTAXE ABSTRAITE NUMÉRO UN:
SPÉCIFICATION DES OBJETS INFORMATIONNELS**

1 Domaine d'application

La présente Recommandation | Norme internationale participe à la définition de la notation ASN.1 et établit une notation permettant de spécifier les classes d'objets informationnels, les objets informationnels et les ensembles d'objets informationnels.

2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT tient à jour une liste des Recommandations de l'UIT-T en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- Recommandation UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes.*
- Recommandation UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un.*

3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale les définitions suivantes s'appliquent.

3.1 Spécification de la notation de base

La présente Recommandation | Norme internationale utilise les termes définis dans la Rec. UIT-T X.680 | ISO/CEI 8824 1.

3.2 Spécification des contraintes

La présente Recommandation | Norme internationale utilise le terme suivant défini dans la Rec. UIT-T X.682 | ISO/CEI 8824-3:

- contrainte tabulaire.

3.3 Paramétrage des spécifications ASN.1

La présente Recommandation | Norme internationale utilise les termes suivants définis dans la Rec. UIT-T X.683 | ISO/CEI 8824-4:

- a) type paramétré;
- b) valeur paramétrée.

3.4 Autres définitions

3.4.1 table associée: (pour un objet informationnel ou un ensemble d'objets informationnels): table abstraite, dérivable de l'objet ou de l'ensemble d'objets en aplatissant la structure hiérarchique résultant de la présence de champs de liaison (voir 3.4.14).

NOTE – Une table associée peut servir à déterminer la nature précise d'une contrainte donnée (voir la Rec. UIT-T X.682 | ISO/CEI 8824-3), appliquée à l'aide d'un ensemble d'objets.

3.4.2 syntaxe par défaut: la notation qui sera utilisée pour définir les objets informationnels des classes pour lesquelles les spécificateurs n'ont pas établi de syntaxe conviviale (voir 11.9).

3.4.3 syntaxe conviviale: notation, indiquée lors de la définition d'une classe, permettant d'en définir les objets informationnels d'une manière conviviale.

NOTE – Par exemple, la syntaxe conviviale de la classe «OPERATION» pourrait permettre de définir les instances de cette classe à l'aide du mot «ARGUMENT» suivi de la spécification «&ArgumentType», puis du mot «RESULT» suivi de la spécification «&ResultType», et enfin du mot «CODE» suivi de la spécification «&operationCode» (voir 11.10).

3.4.4 ensemble d'objets extensible: ensemble d'objets comportant un marqueur d'extension.

3.4.5 champ: composant d'une classe d'objets informationnels. Chaque champ est un champ de type, un champ de valeur de type fixé, un champ de valeur de type variable, un champ d'ensemble de valeurs de type fixé, un champ d'ensemble de valeurs de type variable, un champ d'objet informationnel ou un champ d'ensemble d'objets informationnels.

3.4.6 nom de champ: nom identifiant un champ d'une classe donnée; il s'agira soit de la classe à laquelle appartient le champ concerné, auquel cas le nom est un nom de champ primitif, soit d'une classe comportant une chaîne de champs de liaison menant à la classe à laquelle appartient le champ concerné (voir 9.13 et 9.14).

3.4.7 champ identificateur: champ de valeur de type fixé d'une classe, choisi pour identifier de manière unique les objets informationnels de cette classe. Les valeurs du champ identificateur, s'il existe, doivent être non ambiguës dans tout ensemble d'objets informationnels défini pour cette classe. Elles peuvent, mais pas nécessairement, servir à identifier de manière non ambiguë les objets informationnels de cette classe dans un cadre plus large.

NOTE 1 – Le champ identificateur a un type ASN.1 fixé et les valeurs de ce type peuvent être acheminées par un protocole afin d'identifier les objets informationnels de cette classe.

NOTE 2 – Le cadre dans lequel l'identificateur est non ambigu est celui d'un ensemble d'objets informationnels. Il pourrait cependant être rendu non ambigu dans le cadre d'une syntaxe abstraite quelconque, ou dans le cadre d'un contexte d'application tout entier, et même dans un cadre global recouvrant toutes les classes et tous les contextes d'application si on utilise le type identificateur d'objet «OBJECT IDENTIFIER» comme champ identificateur.

3.4.8 objet informationnel: instance d'une classe d'objets informationnels, composée d'un ensemble de champs respectant les spécifications des champs de cette classe.

NOTE – Par exemple, une instance particulière de la classe d'objets informationnels «OPERATION» (voir l'exemple du 3.4.9) pourrait être «inversionDeMatrice», avec un champ «&ArgumentType» de type «Matrice», un champ «&ResultType» également de type «Matrice» et un champ «&operationCode» contenant la valeur 7 (voir 10.13).

3.4.9 classe d'objets informationnels (classe): ensemble de champs, formant un modèle pour définir une collection virtuellement illimitée d'objets informationnels qui sont les instances de cette classe.

NOTE – Par exemple, la classe d'objets informationnels «OPERATION» pourrait être définie afin de correspondre au concept «opération» des opérations distantes «ROS». Chacune des diverses spécifications de champs nommés correspondrait alors à un aspect pouvant varier d'une instance d'opération à une autre. On pourrait ainsi avoir trois champs «&ArgumentType», «&ResultType» et «&operationCode», les deux premiers spécifiant des champs de type, le troisième un champ de valeur.

3.4.10 champ d'objet informationnel: champ contenant un objet informationnel d'une classe donnée.

3.4.11 ensemble d'objets informationnels: ensemble non vide d'objets informationnels appartenant tous à la même classe.

NOTE – On peut constituer par exemple l'ensemble d'objets informationnels «OperationsMatricielles» de la classe «OPERATION» (voir l'exemple du 3.4.9) contenant «inversionDeMatrice» (voir l'exemple du 3.4.8) et d'autres opérations similaires telles que «additionDeMatrices», «multiplicationDeMatrices», etc. Un tel ensemble d'objets pourrait servir à définir une syntaxe abstraite comportant les mécanismes d'appel et d'indication des résultats pour toutes ces opérations (voir 12.7).

3.4.12 champ d'ensemble d'objets informationnels: champ contenant un ensemble d'objets informationnels d'une classe donnée.

3.4.13 type instance-de: type défini par référence à une classe d'objets informationnels associant des identificateurs d'objets à des types.

3.4.14 champ de liaison: champ d'objet ou d'ensemble d'objets.

3.4.15 type de champ de classe d'objets: type spécifié par référence à un champ d'une classe d'objets informationnels. Dans la Rec. UIT-T X.682 | ISO/CEI 8824-3, une notation est prévue pour permettre de restreindre ce type en faisant référence à un ensemble d'objets informationnels de cette classe.

3.4.16 nom de champ primitif: nom directement spécifié dans une définition de classe d'objets informationnels sans utilisation d'un champ de liaison.

3.4.17 champ de type: champ contenant un type quelconque.

3.4.18 champ de valeur: champ contenant une valeur. Ce champ est soit de type fixé, soit de type variable. Dans le premier cas, le type de la valeur est fixé par la spécification du champ. Dans le second, il est contenu dans un champ de type (spécifique) du même objet informationnel.

3.4.19 champ d'ensemble de valeurs: champ contenant un ensemble non vide de valeurs d'un type donné. Ce champ est soit de type fixé, soit de type variable. Dans le premier cas, le type des valeurs est fixé par la spécification du champ. Dans le second, il est contenu dans un champ de type (spécifique) du même objet informationnel.

NOTE – L'ensemble de valeurs d'un champ d'ensemble de valeurs d'un objet informationnel est un sous-type du type spécifié.

4 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale l'abréviation suivante est utilisée:

ASN.1 Notation de syntaxe abstraite numéro un (*abstract syntax notation one*)

5 Conventions

La présente Recommandation | Norme internationale utilise les conventions de notation définies à l'article 5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

6 Notation

Cet article récapitule la notation définie dans la présente Recommandation | Norme internationale.

6.1 Affectations

La présente Recommandation | Norme internationale définit les notations suivantes, qui peuvent être utilisées comme formes possibles d'affectation «Assignment» (voir l'article 12 de la Rec. UIT-T X.680 | ISO/CEI 8824-1):

- ObjectClassAssignment (affectation de classe d'objets) (voir 9.1);
- ObjectAssignment (affectation d'objet) (voir 11.1);
- ObjectSetAssignment (affectation d'ensemble d'objets) (voir 12.1).

6.2 Types

6.2.1 La présente Recommandation | Norme internationale définit les notations suivantes, qui peuvent être utilisées comme formes possibles du type prédéfini «BuiltinType» (voir 16.2 de la Rec. UIT-T X.680 | ISO/CEI 8824-1):

- ObjectClassFieldType (type de champ de classe d'objets) (voir 14.1);
- InstanceOfType (type instance-de) (voir l'Annexe C).

6.2.2 La présente Recommandation | Norme internationale définit les notations suivantes, qui peuvent être utilisées comme formes possibles d'une référence de type «ReferencedType» (voir 16.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1):

- TypeFromObject (type extrait d'un objet) (voir l'article 15);
- ValueSetFromObjects (ensemble de valeurs extraites d'objets) (voir l'article 15).

6.3 Valeurs

6.3.1 La présente Recommandation | Norme internationale définit les notations suivantes, qui peuvent être utilisées comme formes possibles de valeur prédéfinie «BuiltinValue» (voir 16.8 de la Rec. UIT-T X.680 | ISO/CEI 8824-1):

- ObjectClassFieldValue (valeur de champ de classe d'objets) (voir 14.6);
- InstanceOfValue (valeur instance de) (voir l'Annexe C).

6.3.2 La présente Recommandation | Norme internationale définit la notation suivante, qui peut être utilisée comme forme possible d'une référence de valeur «ReferencedValue» (voir 16.9 de la Rec. UIT-T X.680 | ISO/CEI 8824-1):

- ValueFromObject (valeur extraite d'un objet) (voir l'article 15).

6.4 Eléments

6.4.1 La présente Recommandation | Norme internationale définit la notation suivante, qui peut être utilisée comme forme possible d'élément «Elements» (voir 46.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1):

- ObjectSetElements (éléments ensemble d'objets) (voir 12.6).

7 Unités lexicales ASN.1

Outre les unités lexicales ASN.1 spécifiées à l'article 11 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, la présente Recommandation | Norme internationale utilise les unités lexicales ASN.1 spécifiées dans les paragraphes suivants. Les règles générales applicables à ces unités lexicales sont celles qui sont définies au paragraphe 11.1 de la Rec. UIT-T X.680 | ISO/CEI 8824-1. Ces nouvelles unités lexicales ASN.1 utilisent le jeu de caractères ASN.1 spécifié à l'article 10 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, plus le caractère esperluette «&».

NOTE – La Note du paragraphe 10.1 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 s'applique aussi aux unités lexicales spécifiées aux 7.1 à 7.9 ci-dessous.

7.1 Référence de classe d'objets informationnels

Nom de l'unité lexicale – objectclassreference

Une référence de classe d'objets informationnels «objectclassreference» est composée d'une séquence de caractères conforme à ce qui a été spécifié pour la référence de type «typereference» au paragraphe 11.2 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, sauf qu'elle ne doit pas comporter de minuscules.

7.2 Référence d'objets informationnels

Nom de l'unité lexicale – objectreference

Une référence d'objets informationnels «objectreference» est composée d'une séquence de caractères conforme à ce qui a été spécifié pour la référence de valeur «valuereference» au paragraphe 11.4 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

7.3 Référence d'ensemble d'objets

Nom de l'unité lexicale – objectsetreference

Une référence d'ensemble d'objets «objectsetreference» est composée d'une séquence de caractères conforme à ce qui a été spécifié pour la référence de type «typereference» au paragraphe 11.2 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

7.4 Référence de champ de type

Nom de l'unité lexicale – typefieldreference

Une référence de champ de type «typefieldreference» est composée d'une esperluette «&» immédiatement suivie d'une séquence de caractères conforme à ce qui a été spécifié pour la référence de type «typereference» au paragraphe 11.2 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

7.5 Référence de champ de valeur

Nom de l'unité lexicale – valuefieldreference

Une référence de champ de valeur «valuefieldreference» est composée d'une esperluette «&» immédiatement suivie d'une séquence de caractères conforme à ce qui a été spécifié pour la référence de valeur «valuereference» au paragraphe 11.4 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

7.6 Référence de champ d'ensemble de valeurs

Nom de l'unité lexicale – valuesetfieldreference

Une référence de champ d'ensemble de valeurs «valuesetfieldreference» est composée d'une esperluette «&» immédiatement suivie d'une séquence de caractères conforme à ce qui a été spécifié pour la référence de type «typereference» au paragraphe 11.2 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

7.7 Référence de champ d'objet

Nom de l'unité lexicale – objectfieldreference

Une référence de champ d'objet «objectfieldreference» est composée d'une esperluette «&» immédiatement suivie d'une séquence de caractères conforme à ce qui a été spécifié pour la référence d'objet «objectreference» au 7.2.

7.8 Référence de champ d'ensemble d'objets

Nom de l'unité lexicale – objectsetfieldreference

Une référence de champ d'ensemble d'objets «objectsetfieldreference» est composée d'une esperluette «&» immédiatement suivie d'une séquence de caractères conforme à ce qui a été spécifié pour la référence d'ensemble d'objets «objectsetreference» au 7.3.

7.9 Mot

Nom de l'unité lexicale – word

Un mot «word» est composé d'une séquence de caractères conforme à ce qui a été spécifié pour une référence de type «typereference» au 11.2 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, mais ne doit pas comporter de minuscules ni de chiffres.

7.10 Autres unités lexicales mots clés

Les noms des unités lexicales: «CLASS», «INSTANCE», «SYNTAX», «UNIQUE» sont cités au paragraphe 11.18 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 parmi les mots réservés de la notation ASN.1.

8 Définitions de références

8.1 Les structures:

```
DefinedObjectClass ::=
  ExternalObjectClassReference |
  objectclassreference |
  UsefulObjectClassReference
```

```
DefinedObject ::=
  ExternalObjectReference |
  objectreference
```

```
DefinedObjectSet ::=
  ExternalObjectSetReference |
  objectsetreference
```

sont utilisées pour faire référence à des définitions respectivement de classe, d'objet informationnel et d'ensemble d'objets informationnels.

8.2 Sauf spécification donnée au 12.15 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, les formes «objectclassreference», «objectreference» et «objectsetreference» ne seront utilisées que dans le module dans lequel une classe, un objet informationnel ou un ensemble d'objets informationnels est affecté à cette référence (voir 9.1, 11.1 et 12.1).

Les formes «ExternalObjectClassReference», «ExternalObjectReference» et «ExternalObjectSetReference» sont définies comme suit:

```
ExternalObjectClassReference ::=
  modulereference
  "."
  objectclassreference
```

```
ExternalObjectReference ::=
  modulereference
  "."
  objectreference
```

```
ExternalObjectSetReference ::=
  modulereference
  "."
  objectsetreference
```

Ces formes ne seront pas utilisées, sauf si la référence «objectclassreference», «objectreference» ou «objectsetreference» correspondante s'est vue affecter respectivement une classe, un objet informationnel ou un ensemble d'objets informationnels (voir 9.1, 11.1 et 12.1) dans le module (distinct du module qui y fait référence) identifié par «modulereference». C'est alors respectivement cette classe, cet objet informationnel ou cet ensemble d'objets informationnels qui est désigné par la référence.

8.3 La forme «Usefulobjectclassreference» de «DefinedObjectClass» est définie comme suit:

```
UsefulObjectClassReference ::= TYPE-IDENTIFIER | ABSTRACT-SYNTAX
```

la première forme possible étant spécifiée dans l'Annexe A, la seconde dans l'Annexe B.

NOTE – Les noms des unités lexicales «TYPE-IDENTIFIER» et «ABSTRACT-SYNTAX» sont cités au paragraphe 11.18 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 comme mots réservés de la notation ASN.1.

9 Définition et affectation d'une classe d'objets informationnels

9.1 La notation «ObjectClassAssignment» (affectation de classe d'objets) est utilisée pour affecter une classe d'objets informationnels à un nom de référence «objectclassreference». Cette notation, qui est l'une des formes possibles d'affectation «Assignment» décrite à l'article 12 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, est définie comme suit:

```
ObjectClassAssignment ::=
  objectclassreference
  "::~"
  ObjectClass
```

9.2 La classe d'objets informationnels est celle qui est définie par la structure «ObjectClass» (classe d'objets):

```
ObjectClass ::=
  DefinedObjectClass |
  ObjectClassDefn |
  ParameterizedObjectClass
```

Si la classe d'objets «ObjectClass» est:

- a) une classe d'objets définie «DefinedObjectClass», la définition de la classe est alors la même que celle de la classe donnée en référence;
- b) une définition de classe d'objets «ObjectClassDefn», la classe est alors définie selon les dispositions du 9.3;
- c) une classe d'objets paramétrée «ParameterizedObjectClass», la classe est alors définie selon les dispositions du paragraphe 9.2 de la Rec. UIT-T X.683 | ISO/CEI 8824-4.

9.3 Chaque classe est définie en dernier lieu par une définition de classe d'objets «ObjectClassDefn»:

```
ObjectClassDefn ::=
  CLASS
  "{" FieldSpec "," + "}"
  WithSyntaxSpec?

WithSyntaxSpec ::= WITH SYNTAX SyntaxList
```

Cette notation permet, lors de la définition d'une classe d'indiquer les spécifications des champs nommés, chacune d'elles étant une spécification de champ «FieldSpec» telle que définie au 9.4. A titre optionnel, l'auteur de la spécification peut fournir une syntaxe conviviale d'aide à la définition des objets informationnels «SyntaxList» telle que celle-ci est définie au 10.5. L'auteur de la spécification peut aussi préciser la sémantique associée à la définition de cette classe.

9.4 Chaque spécification de champ «FieldSpec» définit et nomme l'un des champs qui devra ou pourra être associé aux instances de la classe:

```
FieldSpec ::=
  TypeFieldSpec |
  FixedTypeValueFieldSpec |
  VariableTypeValueFieldSpec |
  FixedTypeValueSetFieldSpec |
  VariableTypeValueSetFieldSpec |
  ObjectFieldSpec |
  ObjectSetFieldSpec
```

Les différentes formes possibles de «FieldSpec» sont spécifiées dans les paragraphes suivants.

9.5 La production «TypeFieldSpec» spécifie un champ de type (voir 3.4.17):

```
TypeFieldSpec ::=
  typefieldreference
  TypeOptionalitySpec?

TypeOptionalitySpec ::= OPTIONAL | DEFAULT Type
```

Le nom du champ est «typefieldreference» (référence de champ de type). Si la spécification «TypeOptionalitySpec» est absente, toute définition d'objet informationnel de la classe devra comporter une spécification de type pour ce champ. Si la déclaration «OPTIONAL» est présente, le champ peut alors être laissé non défini. Si la déclaration «DEFAULT» est présente, le «Type» indiqué à la suite indiquera la nature du champ si celui-ci est omis dans la définition.

9.6 La production «FixedTypeValueFieldSpec» spécifie un champ de valeur d'un type fixé (voir 3.4.18):

```
FixedTypeValueFieldSpec ::=
  valuefieldreference
  Type
  UNIQUE?
  ValueOptionalitySpec?

ValueOptionalitySpec ::= OPTIONAL | DEFAULT Value
```

Le nom du champ est «valuefieldreference» (référence de champ de valeur). La notation «Type» spécifie le type de la valeur contenue dans le champ. La déclaration «ValueOptionalitySpec», si elle est présente, spécifie que la valeur peut être omise dans une définition d'objet informationnel, et, dans le cas «DEFAULT», que cette omission entraînera l'affectation de la valeur «Value» indiquée à la suite, qui doit être de ce type. La présence du mot clé «UNIQUE» spécifie que ce champ est un champ identificateur. Si ce mot clé est présent, la spécification «ValueOptionalitySpec» ne prendra pas la forme «DEFAULT Value» (valeur par défaut).

9.7 Lorsqu'une valeur est affectée à un champ identificateur, elle doit être non ambiguë dans le cadre de tout ensemble d'objets informationnels défini.

9.8 La production «VariableTypeValueFieldSpec» spécifie un champ de valeur de type variable (voir 3.4.18):

VariableTypeValueFieldSpec ::=
valuefieldreference
FieldName
ValueOptionalitySpec?

Le nom de ce champ est «valuefieldreference». Le nom de champ «FieldName» (voir 9.14) qui se rapporte à la classe en cours de spécification sera celui d'un champ de type; ce champ de type, qui appartient au même objet informationnel que le champ de valeur, ou qui y est lié par une chaîne de champs d'objet dont les références figureront dans le nom de champ «FieldName», contiendra le type de la valeur. (Tous les champs de liaison dont les références de champ figurent dans le nom de champ «FieldName» sont des champs d'objet.) La déclaration «ValueOptionalitySpec», si elle est présente, spécifie que la valeur peut être omise dans une définition d'objet informationnel, et, dans le cas «DEFAULT», que cette omission entraînera l'affectation de la valeur «Value» indiquée à la suite. La déclaration «ValueOptionalitySpec» sera telle que:

- a) si le champ de type désigné par le nom de champ «FieldName» est déclaré dans une notation «TypeOptionalitySpec» comme étant optionnel «OPTIONAL», la déclaration «ValueOptionalitySpec» aura alors aussi la forme «OPTIONAL»;
- b) si la spécification «ValueOptionalitySpec» est «DEFAULT Value» (valeur par défaut), le champ de type désigné par le nom de champ «FieldName» aura alors une spécification «TypeOptionalitySpec» de «DEFAULT Type» (type par défaut), et la valeur «Value» appartiendra à ce «Type».

9.9 La production «FixedTypeValueSetFieldSpec» spécifie un champ d'ensemble de valeurs de type fixé (voir 3.4.19):

FixedTypeValueSetFieldSpec ::=
valuesetfieldreference
Type
ValueSetOptionalitySpec?

ValueSetOptionalitySpec ::= OPTIONAL | DEFAULT ValueSet

NOTE – La notation «ValueSet» (ensemble de valeurs) est définie dans la Rec. UIT-T X.680 | ISO/CEI 8824-1, paragraphes 15.4 et 15.5 et permet d'énumérer explicitement (entre accolades) l'ensemble des valeurs, ou d'utiliser une référence de type «typereference» pour un sous-type de «Type».

Le nom de ce champ est «valuesetfieldreference» (référence de champ d'ensemble de valeurs). La notation «Type» spécifie le type des valeurs contenues dans ce champ. La déclaration «ValueSetOptionalitySpec», si elle est présente, spécifie que le champ peut être omis dans une définition d'objet informationnel, et, dans le cas «DEFAULT», que son omission entraînera l'affectation de l'ensemble de valeurs «ValueSet» indiqué à la suite, qui doit être un sous-type de «Type».

9.10 La production «VariableTypeValueSetFieldSpec» spécifie un champ d'ensemble de valeurs de type variable (voir 3.4.19):

VariableTypeValueSetFieldSpec ::=
valuesetfieldreference
FieldName
ValueSetOptionalitySpec?

Le nom de ce champ est «valuesetfieldreference». Le nom de champ «FieldName» (voir 9.14) qui se rapporte à la classe en cours de définition est celui d'un champ de type; ce champ de type, qui appartient au même objet informationnel que le champ d'ensemble de valeurs, ou qui y est lié par la chaîne de champs d'objet dont les références figureront dans le nom de champ «FieldName», contiendra le type de la valeur. (Tous les champs de liaison dont les références de champ

figurent dans le nom de champ «FieldName» sont des champs d'objet.) La déclaration «ValueSetOptionalitySpec», si elle est présente, spécifie que l'ensemble de valeurs peut être omis dans une définition d'objet informationnel, et, dans le cas «DEFAULT», que son omission entraînera l'affectation de l'ensemble de valeurs «ValueSet» indiqué à la suite. La déclaration «ValueSetOptionalitySpec» sera telle que:

- a) si le champ de type désigné par le nom de champ «FieldName» est déclaré dans une notation «TypeOptionalitySpec» comme étant optionnel «OPTIONAL», la déclaration «ValueSetOptionalitySpec» aura alors aussi la forme «OPTIONAL»;
- b) si la spécification «ValueSetOptionalitySpec» est «DEFAULT ValueSet» (ensemble de valeurs par défaut), le champ de type désigné par le nom de champ «FieldName» aura alors une spécification «TypeOptionalitySpec» de «DEFAULT Type» (type par défaut), et l'ensemble de valeurs «ValueSet» sera un sous-type de ce «Type».

9.11 La production «ObjectFieldSpec» spécifie un champ d'objet informationnel (voir 3.4.10):

```
ObjectFieldSpec ::=
  objectfieldreference
  DefinedObjectClass
  ObjectOptionalitySpec?
```

ObjectOptionalitySpec ::= OPTIONAL | DEFAULT Object

Le nom de ce champ est «objectfieldreference» (référence de champ d'objet). La notation «DefinedObjectClass» indique la classe de l'objet contenu dans le champ (qui peut être la classe d'objet «ObjectClass» en cours de définition). La déclaration «ObjectOptionalitySpec», si elle est présente, spécifie que le champ peut être omis dans une définition d'objet informationnel, et, dans le cas «DEFAULT», que son omission entraînera l'affectation de l'objet «Object» indiqué à la suite (voir 11.2) qui devra appartenir à la classe «DefinedObjectClass».

9.12 La production «ObjectSetFieldSpec» spécifie un champ d'ensemble d'objets informationnels (voir 3.4.12):

```
ObjectSetFieldSpec ::=
  objectsetfieldreference
  DefinedObjectClass
  ObjectSetOptionalitySpec?
```

ObjectSetOptionalitySpec ::= OPTIONAL | DEFAULT ObjectSet

Le nom de ce champ est «objectsetfieldreference» (référence de champ d'ensemble d'objets). La notation «DefinedObjectClass» indique la classe des objets contenus dans le champ (qui peut être la classe d'objets «ObjectClass» en cours de définition). La déclaration «ObjectSetOptionalitySpec», si elle est présente, spécifie que le champ peut être omis dans une définition d'objet informationnel, et, dans le cas «DEFAULT», que son omission entraîne l'affectation de l'ensemble d'objets «ObjectSet» indiqué à la suite (voir 12.2) dont tous les objets devront appartenir à la classe «DefinedObjectClass».

9.13 La structure «PrimitiveFieldName» (nom de champ primitif) est utilisée pour identifier un champ par rapport à la classe contenant sa spécification:

```
PrimitiveFieldName ::=
  typefieldreference |
  valuefieldreference |
  valuesetfieldreference |
  objectfieldreference |
  objectsetfieldreference
```

Les noms de tous les champs spécifiés dans la définition de classe seront distincts.

9.14 La construction «FieldName» (nom de champ) est utilisée pour identifier un champ par rapport à une classe; elle contient soit directement la spécification du champ, soit une chaîne de champs de liaison conduisant à la classe contenant le champ. Cette chaîne se présente sous la forme d'une liste de noms de champs primitifs «PrimitiveFieldName» séparés par des points «.»

FieldName ::= PrimitiveFieldName "." +

9.15 S'il existe une chaîne (de longueur 1 ou plus) de spécifications de champs de liaison (voir 3.4.14) telle que:

- a) le premier champ appartient à la classe en cours de définition;
- b) chacun des champs suivants appartient à la classe utilisée pour définir la précédente;

- c) le dernier champ est défini à l'aide de la classe en cours de définition,

il y aura alors au moins une spécification de champ comportant la déclaration «ObjectOptionalitySpec» ou «ObjectSetOptionalitySpec» (selon le cas).

NOTE – Il s'agit d'empêcher que la définition d'une classe d'objets informationnels récursive ne génère une représentation non finie des objets informationnels de cette classe.

9.16 Exemples

Une version étendue de la classe d'objets informationnels décrite de manière informelle en exemple au 3.4.9 pourrait être définie comme suit:

```
OPERATION ::= CLASS
{
    &ArgumentTypeOPTIONAL,
    &ResultType    OPTIONAL,
    &Errors          ERROR OPTIONAL,
    &Linked          OPERATION OPTIONAL,
    &resultReturnedBOOLEAN DEFAULT TRUE,
    &code           INTEGER UNIQUE
}

ERROR ::= CLASS
{
    &ParameterType    OPTIONAL,
    &code             INTEGER UNIQUE
}
}
```

NOTE 1 – Cet exemple repose sur les concepts (simplifiés dans ce cadre précis) d'opération et d'erreur de la norme sur les opérations distantes «ROS».

NOTE 2 – Les champs spécifiés pour cette classe comportent deux champs de type («&ArgumentType» et «&ResultType»), deux champs d'ensemble d'objets («&Errors» et «&Linked») et deux champs de valeur («&resultReturned» et «&code»), le dernier étant un champ identificateur.

NOTE 3 – Tout ensemble d'objets informationnels composé d'opérations de la classe «OPERATION» doit être tel qu'aucune valeur de champ «&code» ne soit partagée par deux objets de cet ensemble. (La même règle s'applique aux ensembles d'objets de la classe «ERROR».)

NOTE 4 – La classe d'objets informationnels «OPERATION» comprend une chaîne de champs de liaison selon la description du 9.15 ci-dessus. Cette chaîne, qui est de longueur 1, est formée du champ «&Linked», spécifié (récursivement) à l'aide de la classe «OPERATION». Cependant, cette écriture est tout à fait valide car le champ est déclaré optionnel «OPTIONAL».

NOTE 5 – Aucun de ces exemples ne comporte de déclaration «WithSyntaxSpec», mais de tels exemples en seront donnés au 10.13.

10 Syntaxe conviviale

10.1 Il est fréquent qu'une spécification définisse une classe d'objets informationnels, et que de nombreuses autres spécifications en définissent indépendamment les objets informationnels. L'auteur de la spécification peut juger opportun de fournir une notation facile à utiliser pour définir des objets informationnels d'une telle classe.

NOTE – Telle était la principale utilisation de l'ancienne notation de «macro» ASN.1, avant qu'elle ne soit remplacée par le texte de la présente Recommandation | Norme internationale.

10.2 Cet article spécifie une notation permettant à l'auteur d'une spécification, lors de la définition d'une classe d'objets informationnels d'y associer une syntaxe de spécification des objets informationnels propre à cette classe.

10.3 Cette notation est la structure syntaxique «SyntaxList», qui apparaît dans la structure syntaxique «ObjectClassDefn» (définition de classe d'objets).

10.4 Une structure «SyntaxList» spécifie la syntaxe servant à définir un objet informationnel simple de la classe en cours de définition. Cette syntaxe apparaît comme la syntaxe conviviale «DefinedSyntax» dans les articles suivants.

NOTE – C'est une propriété de la présente Recommandation | Norme internationale de pouvoir déterminer la fin d'une structure syntaxique définie par une liste «SyntaxList» (une instance de «DefinedSyntax»):

- en ignorant les commentaires ASN.1;
- en traitant les valeurs de chaîne de caractères comme des termes lexicaux;
- en recherchant une première accolade ouverte «{», en appariant les accolades imbriquées «{» et «}», et en terminant sur une accolade fermante «}» non appariée.

10.5 La liste «SyntaxList» spécifie la séquence des termes syntaxiques définis «DefinedSyntaxToken» qui doit apparaître dans la syntaxe définie «DefinedSyntax» (voir 11.5).

SyntaxList ::= "{" TokenOrGroupSpec empty + "'"

TokenOrGroupSpec ::= RequiredToken | OptionalGroup

OptionalGroup ::= "[" TokenOrGroupSpec empty + "]"

RequiredToken ::=

Literal |

PrimitiveFieldName

NOTE – La notation de liste syntaxique «SyntaxList» **ne s’est pas** vue conférer la même puissance d’écriture que le formalisme de Backus-Naur (BNF). En résumé, la puissance de cette notation est équivalente à celle que confèrent habituellement les syntaxes de spécification par ligne de commande des interpréteurs de commandes. La liste des termes requis «RequiredToken» possibles est donnée dans l’ordre où ils doivent apparaître; il est possible de rendre optionnels un ou plusieurs termes consécutifs en les plaçant entre crochets.

10.6 Aucun des mots suivants ne doit être utilisé comme littéral «Literal»:

BIT

BOOLEAN

CHARACTER

CHOICE

EMBEDDED

END

ENUMERATED

EXTERNAL

FALSE

INSTANCE

INTEGER

INTERSECTION

MINUS-INFINITY

NULL

OBJECT

OCTET

PLUS-INFINITY

REAL

SEQUENCE

SET

TRUE

UNION

NOTE – Cette liste comprend tous les mots réservés de la notation ASN.1 qui peuvent apparaître comme première unité lexicale d’une production «Type», «Value», «ValueSet», «Object» ou «ObjectSet», plus le mot réservé «END». L’utilisation des autres mots réservés de la notation ASN.1 n’entraîne aucune ambiguïté et est autorisée. Quand la syntaxe conviviale est utilisée dans un environnement dans lequel un mot «word» est aussi une référence de type «typereference» ou d’ensemble d’objets «objectsetreference», son utilisation en tant que mot «word» a priorité.

10.7 Un littéral «Literal» spécifie l’inclusion effective du mot «word» ou de la virgule «,» à cette position dans la syntaxe conviviale:

Literal ::=

word |

","

10.8 Chaque nom de champ primitif «PrimitiveFieldName» spécifie l’insertion (à cette position dans la nouvelle syntaxe) d’une définition «Setting» (voir 11.6) du champ correspondant.

10.9 Chaque nom de champ primitif «PrimitiveFieldName» de la classe d’objets informationnels apparaîtra une fois et une seule.

10.10 Si on rencontre dans le processus d’analyse un groupe optionnel «OptionalGroup», et que l’unité lexicale ASN.1 suivante est syntaxiquement acceptable comme première unité lexicale ASN.1 du groupe optionnel, ce groupe est alors considéré comme présent. Si l’unité lexicale ASN.1 n’est pas syntaxiquement acceptable comme première unité lexicale ASN.1 du groupe optionnel, ce groupe est considéré comme absent.

NOTE – Pour éviter tout effet imprévu, les concepteurs devraient normalement prendre un littéral «Literal» comme première unité lexicale ASN.1 d’un groupe optionnel.

10.11 Une instance de syntaxe définie «DefinedSyntax» n’est pas valide si elle ne spécifie pas tous les champs obligatoires de la classe d’objets informationnels.

10.12 Pour faciliter l'analyse de la nouvelle syntaxe et prévenir tout risque de confusion, la spécification d'une nouvelle syntaxe est soumise aux restrictions supplémentaires suivantes:

- a) chaque groupe «OptionalGroup» doit contenir au moins un nom «PrimitiveFieldName» ou un groupe «OptionalGroup»;
 - NOTE 1 – Cette disposition est destinée à empêcher que ne se constitue un groupe dont aucune information ne se reflète dans un des champs de l'objet informationnel.
- b) l'utilisation des groupes «OptionalGroup» doit être telle qu'à aucun moment du processus d'analyse syntaxique, il ne puisse y avoir ambiguïté sur le nom de champ «FieldName» auquel affecter la définition «Setting»;
- c) si un groupe «OptionalGroup» commence par un littéral «Literal», le premier terme venant après ce groupe doit aussi être un littéral «Literal», différent des premiers littéraux de tous les groupes «OptionalGroup» qui le précèdent immédiatement;

l'utilisation de la syntaxe conviviale «DefinedSyntax» est pour sa part soumise à la restriction suivante:

- d) chaque fois que la syntaxe «DefinedSyntax» d'un groupe «OptionalGroup» comporte un littéral «Literal», elle devra également comporter la définition «Setting» d'un nom «PrimitiveFieldName» de ce groupe.

NOTE 2 – Cette disposition est destinée à empêcher que ne se constitue un groupe dont aucune information ne se reflète dans un des champs de l'objet informationnel.

NOTE 3 – La syntaxe ci-dessous est valide mais la restriction d) empêche l'utilisateur d'écrire «LITERAL» sans le faire suivre d'au moins un des deux groupes optionnels:

[LITERAL [A &field1] [B &field2]]

10.13 Exemples

Les exemples de définitions de classe du 9.16 ci-dessus peuvent être complétées par une syntaxe conviviale servant de méthode pour définir les instances de ces classes. (Cette syntaxe conviviale est utilisée dans l'exemple du 11.10.)

```

OPERATION ::= CLASS
{
    &ArgumentType          OPTIONAL,
    &ResultType            OPTIONAL,
    &Errors                 ERROR OPTIONAL,
    &Linked                OPERATION OPTIONAL,
    &resultReturned        BOOLEAN DEFAULT TRUE,
    &operationCode         INTEGER UNIQUE
}
WITH SYNTAX
{
    [ARGUMENT              &ArgumentType]
    [RESULT                &ResultType]
    [RETURN RESULT        &resultReturned]
    [ERRORS                &Errors]
    [LINKED                &Linked]
    CODE                  &operationCode
}

ERROR ::= CLASS
{
    &ParameterType        OPTIONAL,
    &errorCode             INTEGER UNIQUE
}
WITH SYNTAX
{
    [PARAMETER            &ParameterType]
    CODE                  &errorCode
}
    
```

11 Définition et affectation des objets informationnels

11.1 La production syntaxique «ObjectAssignment» (affectation d'objet) sert à affecter un objet informationnel d'une classe donnée à un nom de référence d'objet «objectreference». Cette production, qui est l'une des formes possibles de la production d'affectation «Assignment» décrite à l'article 12 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, est définie comme suit:

```
ObjectAssignment ::=
  objectreference
  DefinedObjectClass
  "::<="
  Object
```

11.2 L'objet informationnel, qui doit appartenir à la classe indiquée par «DefinedObjectClass», est celui qui est défini par la production «Object» (objet).

```
Object ::=
  DefinedObject |
  ObjectDefn |
  ObjectFromObject |
  ParameterizedObject
```

Si l'objet «Object» est:

- un objet défini «DefinedObject», il est alors identique à celui auquel il est fait référence;
- une définition d'objet «ObjectDefn», il est alors tel que le spécifie le paragraphe 11.3;
- un objet pris dans un objet «ObjectFromObject», il est alors tel que le spécifie l'article 15;
- un objet paramétré «ParameterizedObject», il est alors défini comme selon les dispositions du paragraphe 9.2 de la Rec. UIT-T X.683 | ISO/CEI 8824-4.

11.3 Chaque objet informationnel est défini en dernier lieu par une définition d'objet «ObjectDefn»:

```
ObjectDefn ::=
  DefaultSyntax |
  DefinedSyntax
```

Cette définition «ObjectDefn» sera une syntaxe par défaut «DefaultSyntax» (voir 11.4) si la définition de la classe ne comporte pas de spécification de syntaxe «WithSyntaxSpec», ou une syntaxe conviviale «DefinedSyntax» (voir 11.5) dans le cas contraire.

11.4 La production «DefaultSyntax» est définie comme suit:

```
DefaultSyntax ::= "{" FieldSetting "," * "}"
FieldSetting ::= PrimitiveFieldName Setting
```

Dans la définition de la classe, il y aura une et une seule définition de champ «FieldSetting» pour chacune des spécifications de champ «FieldSpec» qui n'est déclarée ni optionnelle «OPTIONAL» ni à valeur par défaut «DEFAULT», et il y aura au plus une définition de champ «FieldSetting» pour chacune des autres spécifications «FieldSpec». Ces définitions de champ peuvent apparaître dans un ordre quelconque. Le nom de champ primitif «PrimitiveFieldName» de chacune de ces définitions doit être le nom de la spécification de champ correspondante «FieldSpec». La notation de définition «Setting» est spécifiée au 11.6.

11.5 La production «DefinedSyntax» est définie comme suit:

```
DefinedSyntax ::= "{" DefinedSyntaxToken empty * "}"
DefinedSyntaxToken ::=
  Literal |
  Setting
```

La liste syntaxique «SyntaxList» de la spécification «WithSyntaxSpec» (voir l'article 10) détermine la séquence de termes syntaxiques définis «DefinedSyntaxToken» qui doit apparaître dans la syntaxe conviviale «DefinedSyntax». La notation «Setting» est spécifiée au 11.6; chaque instance spécifie la définition d'un champ de l'objet informationnel. La notation «Literal» (littéral) est définie à au 10.7; ces littéraux assurent la lisibilité de la syntaxe par l'homme.

11.6 La notation «Setting» précise le contenu d'un champ dans un objet informationnel en cours de définition:

```
Setting ::=
    Type |
    Value |
    ValueSet |
    Object |
    ObjectSet
```

Si le champ est:

- a) un champ de type, on choisira la forme «Type»;
- b) un champ de valeur, on choisira la forme «Value»;
- c) un champ d'ensemble de valeurs, on choisira la forme «ValueSet»;
- d) un champ d'objet informationnel, on choisira la forme «Object»;
- e) un champ d'ensemble d'objets informationnels, on choisira la forme «ObjectSet».

NOTE – Les valorisations sont soumises aux restrictions supplémentaires indiquées aux 9.5 à 9.12 et aux 11.7 et 11.8.

11.7 Un champ de valeur de type variable sera défini par une valeur du type spécifié par le champ de type correspondant de l'objet lui-même ou de l'objet lié (c'est-à-dire qu'on n'utilisera pas la notation de valeur d'un type ouvert).

11.8 Un champ d'ensemble de valeurs de type variable sera défini par un ensemble de valeurs du type spécifié par le champ de type correspondant de l'objet lui-même ou de l'objet lié (c'est-à-dire qu'on n'utilisera pas la notation de valeur d'un type ouvert).

11.9 Exemples (syntaxe par défaut)

Avec la définition des classes d'objets informationnels donnée au 9.16 ci-dessus (qui ne comportent pas de spécification de syntaxe «WithSyntaxSpec»), les instances de ces classes sont définies à l'aide de la syntaxe par défaut «DefaultSyntax». Par exemple (version étendue de l'exemple donné au 3.4.8):

```
inversionDeMatrice OPERATION ::=
{
    &ArgumentType      Matrice,
    &ResultType         Matrice,
    &Errors              {determinantNul},
    &operationCode      7
}

determinantNul ERROR ::=
{
    &errorCode          1
}
```

11.10 Exemples (syntaxe conviviale)

Au paragraphe 10.13, les exemples de classes sont donnés avec une spécification de syntaxe «WithSyntaxSpec», et les instances de ces classes sont donc définies à l'aide de la syntaxe conviviale «DefinedSyntax». Les exemples du 11.9 s'écriraient donc:

```
inversionDeMatrice OPERATION ::=
{
    ARGUMENT            Matrice
    RESULT              Matrice
    ERRORS              {determinantNul}
    CODE                7
}

determinantNul ERROR ::=
{
    CODE                1
}
```

12 Définition et affectation des ensembles d'objets informationnels

12.1 La production syntaxique «ObjectSetAssignment» affecte un ensemble d'objets informationnels d'une classe spécifiée à un nom de référence d'ensemble d'objets «objectsetreference». Cette production, qui est l'une des formes possibles de la production d'affectation «Assignment» décrite à l'article 12 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, est définie comme suit:

```
ObjectSetAssignment ::=
  objectsetreference
  DefinedObjectClass
  "::<="
  ObjectSet
```

12.2 L'ensemble d'objets informationnels, qui doit appartenir à la classe indiquée par «DefinedObjectClass» (classe d'objets définie), est celui qui est défini par la production «ObjectSet» (ensemble d'objets).

```
ObjectSet ::= "{" ObjectSetSpec "}"
ObjectSetSpec ::= ElementSetSpecs | "..."
```

La production «ElementSetSpecs», spécifiée dans la Rec. UIT-T X.680 | ISO/CEI 8824-1, permet de spécifier un ensemble d'objets informationnels en termes d'objets informationnels ou d'ensembles d'objets informationnels de la classe gouvernante. L'ensemble doit contenir au moins un objet informationnel, sauf si l'ensemble «ObjectSetSpec» est spécifié sous sa deuxième forme. S'il n'en contient pas, les points de suspension indiquent que l'ensemble d'objets est initialement vide, mais que des objets lui seront ajoutés dynamiquement par l'application.

NOTE – Contrairement au type extensible du type ensemble «SET» ou séquence «SEQUENCE», ou aux contraintes extensibles de sous-typage, qui ont un caractère statique par rapport à l'ensemble des valeurs «connues» par une version donnée de la spécification ASN.1, un ensemble d'objets extensible peut s'agrandir et se réduire dynamiquement dans une même version. Il peut même s'agrandir et se réduire dans une même instance d'utilisation d'un programme d'application, par l'addition ou la suppression dynamique de la définition d'un objet.

12.3 Si un ensemble d'objets extensible, A, est référencé dans la définition d'un autre ensemble d'objets B, B hérite de son marqueur d'extension.

12.4 Si un type «ValueSetFromObjects» (voir l'article 15) est défini à l'aide d'un ensemble d'objets extensible, l'ensemble de valeurs résultant n'hérite pas du marqueur d'extension de l'ensemble d'objets.

12.5 Si un type est soumis à une contrainte tabulaire (voir le paragraphe 10.3 de la Rec. UIT-T X.682 | ISO/CEI 8824-3) et que l'ensemble d'objets référencé dans la contrainte tabulaire est extensible, le type n'hérite pas du marqueur d'extension de l'ensemble d'objets. Si le type doit être considéré comme extensible, un marqueur d'extension doit être explicitement ajouté à la construction «ElementSetSpecs».

12.6 La notation «ObjectSetElements» est la suivante:

```
ObjectSetElements ::=
  Object |
  DefinedObjectSet |
  ObjectSetFromObjects |
  ParameterizedObjectSet
```

Les éléments spécifiés par cette notation dépendent de la forme utilisée et sont définis comme suit:

- si c'est la forme «Object» (objet) qui est utilisée, seul l'objet ainsi désigné est alors spécifié. Cet objet doit appartenir à la classe gouvernante;
- si c'est l'une des autres formes qui est utilisée, tous les objets de l'ensemble ainsi désigné sont spécifiés. Ces objets doivent appartenir à la classe gouvernante. Si c'est la forme «DefinedObjectSet» (ensemble d'objets défini) qui est utilisée, l'ensemble d'objets est celui auquel il est fait référence. Si c'est la forme «ObjectSetFromObjects» (ensemble d'objets extraits d'objets) qui est utilisée, l'ensemble d'objets est tel que le spécifie l'article 15. Si c'est la forme «ParameterizedObjectSet» (ensemble d'objets paramétré) qui est utilisée, l'ensemble d'objets est tel que le spécifie le paragraphe 9.2 de la Rec. UIT-T X.683 | ISO/CEI 8824-4.

12.7 Exemple

L'ensemble d'objets informationnels décrit de manière informelle dans la Note du 3.4.11 peut être spécifié comme suit:

```

OperationsMatricielles OPERATION ::=
{
  inversionDeMatrice |
  additionDeMatrices |
  soustractionDeMatrices |
  multiplicationDeMatrices
}

```

13 Tables associées

13.1 Chaque objet informationnel ou ensemble d'objets informationnels peut être vu comme une table: la table associée. Chaque cellule de la table associée est vide ou correspond à la définition d'un champ d'objet informationnel. L'ensemble de colonnes de la table associée est déterminé par la classe à laquelle le ou les objets appartiennent; l'ensemble de lignes, lui, est déterminé par le ou les objets concernés.

13.2 Etant donné la définition d'une classe, l'ensemble des colonnes est déterminé comme suit:

- a) il y a une colonne par champ spécifié dans la définition de la classe. Chacune de ces colonnes porte le nom de champ primitif «PrimitiveFieldName» correspondant;
- b) il y a un ensemble de colonnes supplémentaire par champ de liaison spécifié. Cet ensemble de colonnes est déterminé par l'application de ces règles à la classe gouvernant ce champ de liaison, sauf que leurs noms sont préfixés avec le contenu «PrimitiveFieldName» du champ de liaison suivi d'un point «.».

NOTE – Ces règles sont récursives et sont telles que si une classe fait directement ou non référence à elle-même, l'ensemble de colonnes n'est plus fini. Cette possibilité n'est pas interdite.

13.3 Etant donné un objet informationnel d'une certaine classe, la table associée est celle qui résulterait de l'application des dispositions du 13.4 à l'ensemble d'objets ne contenant que cet objet.

13.4 Etant donné un ensemble d'objets informationnels d'une certaine classe, l'ensemble de lignes de la table associée est celui qui résulterait de l'application de la procédure récursive suivante:

- a) commencer en attribuant une ligne à chaque objet de l'ensemble. Dans chacune de ces lignes, les cellules des colonnes placées sous les rubriques «PrimitiveFieldName» seront remplies avec le champ correspondant de l'objet, toutes les autres cellules restant vides,
- b) pour chaque champ de liaison apparaissant sur une ligne quelconque de l'ensemble:
 - 1) créer la table (subordonnée) associée à la classe désignée par ce champ de liaison, puis
 - 2) remplacer la ligne sur laquelle ce champ de liaison apparaît par une collection de lignes, une pour chaque ligne de la table subordonnée associée. Chacune des lignes de cette collection est la même que celle qui est remplacée, sauf que les cellules de la ligne de la table subordonnée associée servent à remplir les cellules correspondantes jusqu'ici vides, dont le nom de champ «FieldName» est préfixé par le nom de champ primitif «PrimitiveFieldName» du champ de liaison.

NOTE – Ces règles sont récursives et sont telles que si un objet informationnel fait directement ou non référence à lui-même, la procédure se répète indéfiniment. Cette possibilité n'est pas interdite. Dans la pratique, il suffit de connaître le contenu des cases dont les noms ont une longueur finie, et une procédure limitée peut être mise au point à cet effet.

13.5 Exemples de noms de champs «FieldName» valides

Les noms de champ «FieldName» suivants sont des noms valides pour la table associée aux objets et ensembles d'objets de la classe «OPERATION» (définie au 10.13):

```

&ArgumentType
&Errors.&Parameter
&Errors.&errorCode
&Linked.&ArgumentType
&Linked.&Linked.&operationCode
&Linked.&Linked.&Linked.&Linked.&Linked.&Errors.&errorCode

```

La classe «OPERATION» faisant référence à elle-même (par le champ «&Linked»), le nombre de colonnes n'est pas fini.

14 Notation du type champ de classe d'objets (object class field type)

Le type désigné par cette notation dépend de la catégorie du nom du champ concerné. Les paragraphes 14.2 à 14.5 spécifient le type désigné pour les différentes catégories de nom de champ.

14.1 La notation d'un type de champ de classe d'objets (voir 3.4.15) est «ObjectClassFieldType»:

```
ObjectClassFieldType ::=
    DefinedObjectClass
    "."
    FieldName
```

le nom de champ «FieldName» ayant la forme spécifiée au 9.14 relativement à la classe identifiée par la notation «DefinedObjectClass» (classe d'objets définie).

14.2 Pour un champ de type, cette notation définit un type ouvert, c'est-à-dire un type dont l'ensemble des valeurs est l'ensemble de toutes les valeurs possibles pouvant être spécifiées en notation ASN.1. La spécification de contraintes au moyen d'un ensemble d'objets informationnels correspondant (voir la Rec. UIT-T X.682 | ISO/CEI 8824-3) permet de restreindre ce type à un type spécifique. Les restrictions suivantes portant sur l'utilisation de la présente notation s'appliquent lorsque le nom de champ «FieldName» désigne un champ de type:

- a) cette notation ne sera pas utilisée, directement ou non, dans la définition du type d'un champ de valeur ou d'ensemble de valeurs d'une classe d'objets informationnels;
- b) cette notation porte une étiquette indéterminée et ne peut donc pas être utilisée s'il est requis qu'elle porte une étiquette distincte de celle d'un autre type;

NOTE 1 – On peut normalement contourner cette restriction en étiquetant (explicitement) le type.

NOTE 2 – Indépendamment du paragraphe 47.7.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 stipulant que tout élément ajouté conceptuellement à un marqueur d'extension doit avoir une étiquette distincte des étiquettes de tous les types ASN.1 connus, le type ouvert ne devra pas être utilisé là où il sera prescrit d'avoir une étiquette distincte de celles des autres éléments ajoutés conceptuellement.

- c) cette notation ne sera pas étiquetée implicitement;
- d) la valeur affectée à un composant défini de cette manière doit être codée par les règles de codage de telle sorte qu'un destinataire puisse déterminer les valeurs abstraites correspondant à toutes les autres parties de la structure dans laquelle ce composant est inclus sans rien connaître du type effectif de ce composant.

NOTE 4 – Cette structure de «Type» sera généralement contrainte par l'utilisation d'un ensemble d'objets informationnels et de la notation «AtNotation» spécifiée à l'article 10 de la Rec. UIT-T X.682 | ISO/CEI 8824-3. L'attention des utilisateurs de la notation ASN.1 est cependant attirée sur le fait que l'utilisation de cette notation sans application de contrainte peut créer une ambiguïté dans les spécifications de mise en œuvre et doit normalement être évitée.

14.3 Pour un champ de valeur ou d'ensemble de valeurs de type fixé, cette notation désigne le «Type» qui figure dans la spécification de ce champ dans la définition de la classe d'objets informationnels.

14.4 Pour un champ de valeur ou d'ensemble de valeurs de type variable, cette notation définit un type ouvert. Son utilisation est soumise aux mêmes restrictions que celles du 14.2.

14.5 La présente notation n'est pas autorisée si le champ est un champ d'objet ou d'ensemble d'objets.

14.6 La notation définissant une valeur de ce type est «ObjectClassFieldValue» (valeur de champ de classe d'objets):

```
ObjectClassFieldValue ::=
    OpenTypeFieldVal |
    FixedTypeFieldVal

OpenTypeFieldVal ::= Type ":" Value

FixedTypeFieldVal ::= BuiltinValue | ReferencedValue
```

14.7 Pour un champ de valeur ou d'ensemble de valeurs de type fixé dans «ObjectClassFieldType», on utilisera la valeur de champ de type fixé dans «FixedTypeFieldVal», qui sera une valeur du «Type» spécifié dans la définition de la classe d'objets informationnels.

14.8 Pour un champ de type, un champ de valeur ou d'ensemble de valeurs de type variable dans «ObjectClassFieldType», on utilisera la valeur de champ de type ouvert «OpenTypeFieldVal». Le «Type» de cette valeur sera un type ASN.1 quelconque et la valeur «Value» une valeur quelconque de ce type.

14.9 Exemple d'utilisation de type champ de classe d'objets «ObjectClassFieldType»

Les exemples suivants reprennent l'exemple du 11.9 et représentent: (a) un type de champ de classe d'objets «ObjectClassFieldType» possible, (b) le type équivalent (lorsqu'il est utilisé sans contrainte) et (c) la notation d'un exemple de valeur de ce type.

- 1 (a) OPERATION.&operationCode
 (b) INTEGER
 (c) 7
- 2 (a) OPERATION.&ArgumentType
 (b) *type ouvert*
 (c) Matrice:
 {{1, 0, 0, 0},
 {0, 1, 0, 0},
 {0, 0, 1, 0},
 {0, 0, 0, 1}}
- 3 (a) OPERATION.&Linked.&Linked.&Errors.&errorCode
 (b) INTEGER
 (c) 1
- 4 (a) OPERATION.&Linked.&ArgumentType
 (b) *type ouvert*
 (c) UniversalString:{ constanteDePlanck, "et", operateurHamiltonien }

15 Extraction d'information à partir d'objets

15.1 Les informations figurant dans les colonnes de la table associée d'un objet ou d'un ensemble d'objets peuvent être utilisées par les différentes formes de la notation «InformationFromObjects».

```

InformationFromObjects ::=
    ValueFromObject           |
    ValueSetFromObjects       |
    TypeFromObject           |
    ObjectFromObject         |
    ObjectSetFromObjects

ValueFromObject ::=
    ReferencedObjects
    "."
    FieldName

ValueSetFromObjects ::=
    ReferencedObjects
    "."
    FieldName

TypeFromObject ::=
    ReferencedObjects
    "."
    FieldName

ObjectFromObject ::=
    ReferencedObjects
    "."
    FieldName
    
```

ObjectSetFromObjects ::=
ReferencedObjects
 "."
FieldName

ReferencedObjects ::=
DefinedObject | ParameterizedObject |
DefinedObjectSet | ParameterizedObjectSet

15.2 Cette notation désigne le contenu intégral de la colonne indiquée par le champ «FieldName» dans la table associée pour les objets indiqués par «ReferencedObjects».

15.3 Selon la forme de la notation «ReferencedObjects» et selon le nom de champ «FieldName», la présente production représente une valeur, un ensemble de valeurs, un type, un objet ou un ensemble d'objets. Ces cinq cas ont été respectivement représentés par les structures «ValueFromObject», «ValueSetFromObjects», «TypeFromObject», «ObjectFromObject» et «ObjectSetFromObjects». Chacune de ces productions est un cas particulier de la production «InformationFromObjects».

15.4 On peut subdiviser la production «InformationFromObjects» en deux parties. La première partie est obtenue en supprimant le dernier ou l'unique nom de champ primitif «PrimitiveFieldName» et le point «.» qui le précède. Si cette première partie désigne un objet ou un ensemble d'objets, les paragraphes 15.5 à 15.9 s'appliquent. Dans le cas contraire, la notation est illicite. La deuxième partie est le dernier ou l'unique nom «PrimitiveFieldName».

NOTE – (explicative) – Etant donné la définition suivante:
 obj.&a.&b.&c.&d

La première partie de la définition est: obj.&a.&b.&c et la deuxième partie &d.

15.5 Les cinq formes de production sont explicitées dans le Tableau 1. La première colonne indique la nature de la première partie de la notation, et la deuxième colonne la nature de la deuxième partie de la notation telles que ces parties ont été définies au 15.4. La troisième colonne indique celle des cinq formes (définies au 15.3) à laquelle la notation «InformationFromObjects» s'applique le cas échéant.

Tableau 1 – Formes licites de la notation «InformationFromObjects»

Première partie de «InformationFromObjects»	Deuxième partie de «InformationFromObjects»	Production (Elément désigné par la notation)
Objet	champ de valeur d'un type fixé	«ValueFromObject» (valeur extraite d'un objet)
	champ de valeur d'un type variable	«ValueFromObject» (valeur extraite d'un objet)
	champ d'ensemble de valeurs d'un type fixé	«ValueSetFromObjects» (ensemble de valeurs extrait d'un objet)
	champ d'ensemble de valeurs d'un type variable	«ValueSetFromObjects» (ensemble de valeurs extrait d'un objet)
	champ de type	«TypeFromObject» (type extrait d'un objet)
	champ d'objet	«ObjectFromObject» (objet extrait d'un objet)
	champ d'ensemble d'objets	«ObjectSetFromObject» (ensemble d'objets extrait d'un objet)
Ensemble d'objets	champ de valeur d'un type fixé	«ValueSetFromObjects» (ensemble de valeurs extraites d'objets)
	champ de valeur d'un type variable	<i>forme illicite</i>
	champ d'ensemble de valeurs d'un type fixé	«ValueSetFromObjects» (ensemble de valeurs extraites d'objets)
	champ d'ensemble de valeurs d'un type variable	<i>forme illicite</i>
	champ de type	<i>forme illicite</i>
	champ d'objet	«ObjectSetFromObjects» (ensemble d'objets extraits d'objets)
	champ d'ensemble d'objets	«ObjectSetFromObjects» (ensemble d'objets extraits d'objets)

15.6 Lorsque la première partie de la notation désigne un ensemble d'objets et que la dernière partie du nom «PrimitiveFieldName» identifie un champ d'ensemble de valeurs de type fixé, la production «ValueSetFromObjects» correspond à la réunion des ensembles de valeurs sélectionnés.

15.7 Lorsque la première partie de la notation désigne un ensemble d'objets et que la dernière partie du nom «PrimitiveFieldName» identifie un champ d'ensemble d'objets, la production «ObjectSetFromObjects» correspond à la réunion des ensembles d'objets sélectionnés.

15.8 Comme l'indique le Tableau 1, la notation n'est pas licite si la première partie de la notation désigne un ensemble d'objets, et que «PrimitiveFieldName» identifie un champ de valeur ou d'ensemble de valeurs de type variable ou un champ de type.

15.9 L'utilisation de cette notation n'est pas autorisée si toutes les cellules de la colonne désignée sont vides, sauf lorsque la notation est utilisée pour définir directement un champ déclaré optionnel «OPTIONAL» (ou par défaut «DEFAULT») d'un objet informationnel, ce champ prenant alors la valeur vide (ou la valeur par défaut).

15.10 Exemple d'information extraite d'objets

Avec les définitions des exemples des paragraphes 11.9, 11.10 et 12.7, les productions suivantes (dans la colonne de gauche) sont valides et peuvent être utilisées comme des équivalents aux expressions figurant dans la colonne de droite.

"ValueFromObject" (valeur extraite d'un objet)

inversionDeMatrice.&operationCode	7
determinantNul.&errorCode	1

"TypeFromObject" (type extrait d'un objet)

inversionDeMatrice.&ArgumentType	Matrice
----------------------------------	---------

"ValueSetFromObjects" (ensemble de valeurs extraites d'objets)

inversionDeMatrice.&Errors.&errorCode	{ 1 }
OperationsMatricielles.&operationCode	{ 7 et les autres }

"ObjectSetFromObjects" (ensemble d'objets extraits d'objets)

inversionDeMatrice.&Errors	{ determinantNul }
OperationsMatricielles.&Errors	{ determinantNul et les autres }

Annexe A

Classe d'objets informationnels «TYPE-IDENTIFIER» (identificateur de type)

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

A.1 La présente annexe spécifie une classe d'objets informationnels utile, dont la référence de classe est «TYPE-IDENTIFIER» (identificateur de type).

NOTE – Cette classe d'objets informationnels est la classe utile la plus simple; elle comporte seulement deux champs, un champ identificateur du type identificateur d'objet «OBJECT IDENTIFIER» et un champ de type définissant le type ASN.1 servant à acheminer toutes les informations concernant chacun des objets de cette classe. Elle est définie dans la présente Recommandation | Norme internationale en raison de l'utilisation très répandue des objets informationnels de cette forme.

A.2 La classe d'objets informationnels «TYPE-IDENTIFIER» est définie comme suit:

```

TYPE-IDENTIFIER ::= CLASS
{
    &id OBJECT IDENTIFIER UNIQUE,
    &Type
}
WITH SYNTAX {&Type IDENTIFIED BY &id}

```

A.3 Cette classe est définie comme une classe «utile» d'objets informationnels et est disponible dans tout module sans qu'il soit nécessaire de l'importer.

A.4 Exemple

Le corps d'une communication de messagerie MHS peut être défini comme suit:

```

CLASSE-DE-CORPS-DE-MESSAGE ::= TYPE-IDENTIFIER

CorpsFaxG4 CLASSE-DE-CORPS-DE-MESSAGE ::=
    {BIT STRING IDENTIFIED BY {corpsDeMessage 3}}

```

Un concepteur de protocole définirait généralement une composante pour acheminer les instances de la classe «CLASSE-DE-CORPS-DE-MESSAGE» en spécifiant le type «INSTANCE OF CLASSE-DE-CORPS-DE-MESSAGE» défini au C.9.

Annexe B

Définition d'une syntaxe abstraite

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

B.1 Cette annexe spécifie la classe utile d'objets informationnels «ABSTRACT-SYNTAX» destinée à la définition des syntaxes abstraites.

NOTE – Il est recommandé de définir une instance de cette classe chaque fois qu'une syntaxe abstraite est définie comme les valeurs d'un type ASN.1 unique.

B.2 La classe d'objets informationnels «ABSTRACT-SYNTAX» est définie comme suit:

```

ABSTRACT-SYNTAX ::= CLASS
{
  &id           OBJECT IDENTIFIER UNIQUE,
  &Type,
  &property     BIT STRING {handles-invalid-encodings(0)} DEFAULT {}
}
WITH SYNTAX {
  &Type IDENTIFIED BY &id [HAS PROPERTY &property]
}

```

Le champ «&id» de chaque syntaxe abstraite est le nom de la syntaxe abstraite, tandis que le champ «&Type» contient le type ASN.1 simple dont les valeurs composent cette syntaxe abstraite. La propriété «handles-invalid-encodings» indique que les codages non valides ne doivent pas être traités comme des erreurs lors du décodage, et qu'il appartient à l'application de décider de la façon dont doivent être traités lesdits codages non valides.

B.3 Cette classe est définie comme une classe «utile» d'objets informationnels en raison de son utilité générale; elle est disponible dans tous les modules sans qu'il soit nécessaire de l'importer.

B.4 Exemple

Si un type ASN.1, appelé XXX-PDU a été défini, on peut alors spécifier une syntaxe abstraite contenant toutes les valeurs du type XXX-PDU à l'aide de la notation:

```

Syntaxe-Abstraite-xxx ABSTRACT-SYNTAX ::=
{ XXX-PDU IDENTIFIED BY {xxx 5} }

```

On trouvera au C.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 un exemple détaillé d'utilisation de la classe d'objets informationnels «ABSTRACT-SYNTAX».

B.5 Il arrive fréquemment qu'une syntaxe abstraite soit définie sous la forme d'un type paramétré (comme le définit la Rec. UIT-T X.683 | ISO/CEI 8824-4), par exemple avec des paramètres imposant des limites à certaines composantes du protocole. Ces paramètres, soumis aux restrictions spécifiées dans l'article 10 de la Rec. UIT-T X.683 | ISO/CEI 8824-4, peuvent être résolus au moment de la définition de la syntaxe abstraite, ou être acheminés en tant que paramètres de la syntaxe abstraite.

Annexe C

Le type instance-de (instance-of type)

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

C.1 La présente annexe spécifie les notations de type et de valeur des types instance-de (voir 3.4.13) capables d'acheminer une valeur quelconque d'un objet informationnel d'une classe définie comme une classe «TYPE-IDENTIFIER» (identificateur de type) (voir l'Annexe A) en utilisant une affectation de classe d'objets informationnels (la référence de classe d'objets informationnels est spécifiée comme une partie de cette notation).

C.2 La notation instance-de «InstanceOfType» est citée au paragraphe 16.2 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 comme l'une des notations produisant «Type» et est définie comme suit:

InstanceOfType ::= INSTANCE OF DefinedObjectClass

NOTE – On trouvera à l'article 10 de la Rec. UIT-T X.682 | ISO/CEI 8824-3 la spécification d'une méthode permettant de contraindre ce type en lui appliquant une «table de contrainte», limitant ses valeurs à celles qui représentent un ensemble spécifique d'objets informationnels de cette classe.

C.3 La présente notation spécifie un type acheminant le champ «&id» (un identificateur d'objet «OBJECT IDENTIFIER») et une valeur du champ «&Type» à partir d'une instance quelconque de la classe d'objets définie «DefinedObjectClass».

NOTE – Cette structure sera normalement contrainte par un ensemble d'objets, généralement (mais pas nécessairement) un nom de référence muet tel que défini dans les paragraphes 8.3 à 8.11 de la Rec. UIT-T X.683 | ISO/CEI 8824-4, dont le véritable ensemble d'objets est défini ailleurs.

C.4 Tous les types instance-de ont une étiquette de la classe universelle, numéro 8.

NOTE – Il s'agit de la même étiquette universelle que celle du type externe, et l'utilisation du type «instance-de» peut être compatible bit par bit avec le type externe lorsque les règles de codage de base ASN.1 sont utilisées.

C.5 Le type instance-de a un type séquence associé qui forme la base de la notation utilisée pour définir des valeurs et des sous-types du type instance-de.

NOTE – Lorsque ce type est contraint par la notation de contrainte définie dans la Rec. UIT-T X.682 | ISO/CEI 8824-3, le type de séquence associé est aussi contraint. Les contraintes portant sur le type de séquence associé résultant d'une contrainte appliquée au type instance-de sont spécifiées dans l'Annexe A de la Rec. UIT-T X.682 | ISO/CEI 8824-3.

C.6 On suppose que le type de séquence associé est défini dans un environnement dans lequel l'étiquetage est rendu explicite par la déclaration «EXPLICIT TAGS».

C.7 Le type de séquence associé sera:

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

où la notation de la classe d'objets définie <DefinedObjectClass> est remplacée par la notation «DefinedObjectClass» particulière utilisée dans la notation instance-de «InstanceOfType».

C.8 La notation de valeur «InstanceOfValue» d'une notation instance-de «InstanceOfType» doit être celle du type de séquence associé.

InstanceOfValue ::= Value

C.9 Exemple

On peut donner l'exemple suivant, construit à partir de l'exemple donné en A.4:

Le type:

INSTANCE OF CLASSE-DE-CORPS-DE-MESSAGE

a un type séquence associé:

```
SEQUENCE
{
  type-id  CLASSE-DE-CORPS-DE-MESSAGE.&id,
  value    [0] CLASSE-DE-CORPS-DE-MESSAGE.&Type
}
```

On trouvera un exemple de l'application d'une contrainte tabulaire à ce type dans l'Annexe A de la Rec. UIT-T X.682 | ISO/CEI 8824-3.

Annexe D

Exemples

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

D.1 Exemple d'utilisation de la classe «OPERATION» simplifiée

Etant donné la définition simple suivante des classes d'objets informationnelles «OPERATION» et «ERROR»:

```

OPERATION ::= CLASS
{
  &ArgumentType      OPTIONAL,
  &ResultType        OPTIONAL,
  &Errors             ERROR OPTIONAL,
  &Linked            OPERATION OPTIONAL,
  &resultReturned    BOOLEAN DEFAULT TRUE,
  &operationCode     INTEGER UNIQUE
}

WITH SYNTAX
{
  [ARGUMENT          &ArgumentType]
  [RESULT            &ResultType]
  [RETURN RESULT    &resultReturned]
  [ERRORS           &Errors]
  [LINKED           &Linked]
  [CODE             &operationCode]
}

ERROR ::= CLASS
{
  &ParameterType    OPTIONAL,
  &errorCode         INTEGER UNIQUE
}

WITH SYNTAX
{
  [PARAMETER        &ParameterType]
  [CODE             &errorCode]
}

```

nous pouvons définir l'ensemble d'objets suivant pouvant contenir deux objets de la classe «OPERATION»:

```

Mes-Operations OPERATION ::= { operationA | operationB }

operationA OPERATION ::= {
  ARGUMENT          INTEGER
  ERRORS            { { PARAMETER INTEGER CODE 1000 } | { CODE 1001 } }
  CODE              1
}

operationB OPERATION ::= {
  ARGUMENT          IA5String
  RESULT            BOOLEAN
  ERRORS            { { CODE 1002 } | { PARAMETER IA5String CODE 1003 } }
  CODE              2
}

```

L'extraction de l'ensemble des objets de la classe «ERROR» de l'ensemble d'objets ci-dessus peut être réalisée de la manière suivante:

```

Les-Erreurs-De-Mes-Operations ERROR ::= { Mes-Operations.&Errors }

```

L'ensemble d'objets résultant est:

```

Les-Erreurs-De-Mes-Operations ERROR ::= {
  { PARAMETER INTEGER CODE 1000 } |
  { CODE 1001 } |
  { CODE 1002 } |
  { PARAMETER IA5String CODE 1003 }
}
    
```

L'extraction de l'ensemble des codes d'erreurs et des erreurs d'opérations peut être réalisée de la manière suivante:

```

Les-Codes-Des-Erreurs-De-Mes-Operations INTEGER ::= { Mes-Operations.&Errors.&errorCode }
    
```

L'ensemble de valeurs résultant est:

```

Les-Codes-Des-Erreurs-De-Mes-Operations INTEGER ::= { 1000 | 1001 | 1002 | 1003 }
    
```

D.2 Exemple d'utilisation du type champ de classe d'objets «ObjectClassFieldType»

Le type de champ de classe d'objets «ObjectClassFieldType» peut servir à spécifier des types, par exemple:

```

-- Les types «ObjectClassFieldType» sont extraits de cette classe.
-- Seuls les cinq premiers champs peuvent être utilisés en extraction.
    
```

```

EXEMPLE-DE-CLASSE ::= CLASS {
  &ChampType                                OPTIONAL,
  &champValeurDeTypeFixe                    INTEGER OPTIONAL,
  &champValeurDeTypeVariable                &ChampType OPTIONAL,
  &ChampEnsembleDeValeursDeTypeFixe        INTEGER OPTIONAL,
  &ChampEnsembleDeValeursDeTypeVariable    &ChampType OPTIONAL,
  &champObjet                               CLASSE-SIMPLE OPTIONAL,
  &ChampEnsembleDObjets                    CLASSE-SIMPLE OPTIONAL
}

WITH SYNTAX {
  [CHAMP-TYPE                                &ChampType]
  [CHAMP-VALEUR-DE-TYPE-FIXE                &champValeurDeTypeFixe]
  [CHAMP-VALEUR-DE-TYPE-VARIABLE            &champValeurDeTypeVariable]
  [CHAMP-ENSEMBLE-DE-VALEURS-DE-TYPE-FIXE  &ChampEnsembleDeValeursDeTypeFixe]
  [CHAMP-ENSEMBLE-DE-VALEURS-DE-TYPE-VARIABLE &ChampEnsembleDeValeursDeTypeVariable]
  [CHAMP-OBJET                              &champObjet]
  [CHAMP-ENSEMBLE-D-OBJETS                  &ChampEnsembleDObjets]
}

CLASSE-SIMPLE ::= CLASS {
  &valeur INTEGER
}

WITH SYNTAX {
  &valeur
}

-- Ce type contient les composants spécifiés au moyen de la notation «ObjectClassFieldType».
-- Dans le cas des champs de type et des champs de valeurs et d'ensembles de valeurs de type
-- variable, le type de composant résultant est un type ouvert. Dans le cas de valeurs
-- et d'ensembles de valeurs de type fixé, le type de composant résultant est un entier.
-- NOTE – Aucune contrainte n'a été imposée aux occurrences suivantes du type «ObjectClassFieldType»;
-- mais normalement, on en indiquera pour faire référence à un tel type.

ExempleDeType ::= SEQUENCE {
  typeOuvertNumero1    EXEMPLE-DE-CLASSE.&ChampType,
  entierNumero1        EXEMPLE-DE-CLASSE.&champValeurDeTypeFixe,
  typeOuvertNumero2    EXEMPLE-DE-CLASSE.&champValeurDeTypeVariable,
  entierNumero2        EXEMPLE-DE-CLASSE.&ChampEnsembleDeValeursDeTypeFixe,
  typeOuvertNumero3    EXEMPLE-DE-CLASSE.&ChampEnsembleDeValeursDeTypeVariable
}
    
```

```

exempleDeValeur ExempleDeType ::= {
    typeOuvertNumero1          BOOLEAN : TRUE,
    entierNumero1              123,
    typeOuvertNumero2          IA5String : "abcdef",
    entierNumero2              456,
    typeOuvertNumero3          BIT STRING : '01010101'B
}

```

D.3 Illustration de l'utilisation des objets et des ensembles d'objets

Dans ce qui suit, on utilise la classe d'objets définie au D.2 ci-dessus:

```

objetA EXEMPLE-DE-CLASSE ::= {
    CHAMP-VALEUR-DE-TYPE-FIXE          123
    CHAMP-ENSEMBLE-DE-VALEURS-DE-TYPE-FIXE { 1 | 2 | 3 }
    CHAMP-OBJET                        { 1 }
    CHAMP-ENSEMBLE-D-OBJETS            { { 2 } | { 3 } }
}

objetB EXEMPLE-DE-CLASSE ::= {
    CHAMP-TYPE                          IA5String
    CHAMP-VALEUR-DE-TYPE-FIXE          456
    CHAMP-VALEUR-DE-TYPE-VARIABLE      "abc"
    CHAMP-ENSEMBLE-DE-VALEURS-DE-TYPE-VARIABLE { "d" | "e" | "f" }
}
-- L'ensemble d'objets suivant contient deux objets définis et un objet en cours de définition.

EnsembleDObjets EXEMPLE-DE-CLASSE ::= {
    objetA |
    objetB |
    {
        CHAMP-TYPE                      INTEGER
        CHAMP-VALEUR-DE-TYPE-FIXE      789
        CHAMP-ENSEMBLE-DE-VALEURS-DE-TYPE-VARIABLE { 4 | 5 | 6 }
    }
}
-- Les définitions suivantes extraient des informations des objets et de l'ensemble d'objets.

valeurEntiere INTEGER ::= objetA.&champValeurDeTypeFixe
valeurChaine IA5String ::= objetB.&champValeurDeTypeVariable
EnsembleDeValeursEntieresDeLObjetA INTEGER ::= { objetA.&ChampEnsembleDeValeursDeTypeFixe }
EnsembleDeValeursChaine IA5String ::= { objetB.&ChampEnsembleDeValeursDeTypeVariable }
TypeChaine ::= objetB.&ChampType
objetDeLObjetA CLASSE-SIMPLE ::= objetA.&champObjet
EnsembleDObjetsDeLObjetA CLASSE-SIMPLE ::= { objetA.&ChampEnsembleDObjets }
EnsembleDeValeursDUnEnsembleDObjets INTEGER ::= { EnsembleDObjets.&champValeurDeTypeFixe }
EnsembleDEnsemblesDeValeursDUnEnsembleDObjets INTEGER ::=
    { EnsembleDObjets.&ChampEnsembleDeValeursDeTypeFixe }
EnsembleDObjetsDUnEnsembleDObjets CLASSE-SIMPLE ::= { EnsembleDObjets.&champObjet }
EnsembleDEnsemblesDObjetsDUnEnsembleDObjets CLASSE-SIMPLE ::=
    { EnsembleDObjets.&ChampEnsembleDObjets }

```

Annexe E

Annexe didactique sur le modèle ASN.1 d'extension des ensembles d'objets

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

E.1 Ensembles d'objets extensibles

Une spécification ASN.1 peut également définir des ensembles d'objets informationnels et de tels ensembles peuvent être marqués comme extensibles au moyen du marqueur d'extension. L'utilisation du marqueur d'extension avec les ensembles d'objets diffère de son utilisation avec les types, en ce sens qu'il indique souvent qu'une application communicante est nécessaire pour ajouter ou supprimer de façon dynamique des objets de l'ensemble. Une contrainte tabulaire ou relationnelle non satisfaite n'est pas considérée en elle-même comme une erreur si l'ensemble d'objets est extensible. L'absence, dans l'ensemble d'objets, de la valeur du type référencé, n'est pas considérée comme une erreur, mais si cette valeur figure, la contrainte qui lui est imposée doit être satisfaite.

Annexe F

Récapitulation de la notation

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

Les unités lexicales suivantes sont définies à l'article 7:

objectclassreference
objectreference
objectsetreference
typefieldreference
valuefieldreference
valuesetfieldreference
objectsetfieldreference
word
CLASS
INSTANCE
SYNTAX
UNIQUE

Les unités lexicales suivantes sont définies dans la Rec. UIT-T X.680 | ISO/CEI 8824-1 et utilisées dans la présente Recommandation | Norme internationale:

empty
modulereference
"::="
"{"
"}"
" "
" "
" "
" "
"["
"]"
":"
DEFAULT
OF
OPTIONAL
WITH

Les productions suivantes sont définies dans la Rec. UIT-T X.680 | ISO/CEI 8824-1 et utilisées dans la présente Recommandation | Norme internationale:

ElementSetSpec
Type
Value
ValueSet

Les productions suivantes sont définies dans la Rec. UIT-T X.683 | ISO/CEI 8824-4 et utilisées dans la présente Recommandation | Norme internationale:

ParameterizedObjectClass
ParameterizedObjectSet
ParameterizedObject

Les productions suivantes sont définies dans la présente Recommandation | Norme internationale:

DefinedObjectClass ::=
ExternalObjectClassReference | objectclassreference | UsefulObjectClassReference
ExternalObjectClassReference ::= modulereference "." objectclassreference
UsefulObjectClassReference ::=
TYPE-IDENTIFIER |
ABSTRACT-SYNTAX

ObjectClassAssignment ::= objectclassreference "::=" ObjectClass
ObjectClass ::= DefinedObjectClass | ObjectClassDefn | ParameterizedObjectClass
ObjectClassDefn ::= CLASS "{" FieldSpec "," + "}" WithSyntaxSpec?
FieldSpec ::=
 TypeFieldSpec |
 FixedTypeValueFieldSpec |
 VariableTypeValueFieldSpec |
 FixedTypeValueSetFieldSpec |
 VariableTypeValueSetFieldSpec |
 ObjectFieldSpec |
 ObjectSetFieldSpec
PrimitiveFieldName ::=
 typefieldreference |
 valuefieldreference |
 valuesetfieldreference |
 objectfieldreference |
 objectsetfieldreference
FieldName ::= PrimitiveFieldName "." +
TypeFieldSpec ::= typefieldreference TypeOptionalitySpec?
TypeOptionalitySpec ::= OPTIONAL | DEFAULT Type
FixedTypeValueFieldSpec ::= valuefieldreference Type UNIQUE? ValueOptionalitySpec?
ValueOptionalitySpec ::= OPTIONAL | DEFAULT Value
VariableTypeValueFieldSpec ::= valuefieldreference FieldName ValueOptionalitySpec?
FixedTypeValueSetFieldSpec ::= valuesetfieldreference Type ValueSetOptionalitySpec?
ValueSetOptionalitySpec ::= OPTIONAL | DEFAULT ValueSet
VariableTypeValueSetFieldSpec ::= valuesetfieldreference FieldName ValueSetOptionalitySpec?
ObjectFieldSpec ::= objectfieldreference DefinedObjectClass ObjectOptionalitySpec?
ObjectOptionalitySpec ::= OPTIONAL | DEFAULT Object
ObjectSetFieldSpec ::= objectsetfieldreference DefinedObjectClass ObjectSetOptionalitySpec?
ObjectSetOptionalitySpec ::= OPTIONAL | DEFAULT ObjectSet
WithSyntaxSpec ::= WITH SYNTAX SyntaxList
SyntaxList ::= "{" TokenOrGroupSpec empty * "}"
TokenOrGroupSpec ::= RequiredToken | OptionalGroup
OptionalGroup ::= "[" TokenOrGroupSpec empty + "]"
RequiredToken ::= Literal | PrimitiveFieldName
Literal ::= word | ","
DefinedObject ::= ExternalObjectReference | objectreference
ExternalObjectReference ::= modulereference "." objectreference
ObjectAssignment ::= objectreference DefinedObjectClass "::=" Object
Object ::= DefinedObject | ObjectDefn | ObjectFromObject | ParameterizedObject
ObjectDefn ::= DefaultSyntax | DefinedSyntax
DefaultSyntax ::= "{" FieldSetting "," * "}"
FieldSetting ::= PrimitiveFieldName Setting
DefinedSyntax ::= "{" DefinedSyntaxToken empty * "}"
DefinedSyntaxToken ::= Literal | Setting

Setting ::= Type | Value | ValueSet | Object | ObjectSet
DefinedObjectSet ::= ExternalObjectSetReference | objectsetreference
ExternalObjectSetReference ::= modulereference "." objectsetreference
ObjectSetAssignment ::= objectsetreference DefinedObjectClass "==" ObjectSet
ObjectSet ::= "{" ObjectSetSpec "}"
ObjectSetSpec ::= ElementSetSpecs | "..."
ObjectSetElements ::=
 Object | DefinedObjectSet | ObjectSetFromObjects | ParameterizedObjectSet
ObjectClassFieldType ::= DefinedObjectClass "." FieldName
ObjectClassFieldValue ::= OpenTypeFieldVal | FixedTypeFieldVal
OpenTypeFieldVal ::= Type ":" Value
FixedTypeFieldVal ::= Value
InformationFromObjects ::= ValueFromObject | ValueSetFromObjects |
 TypeFromObject | ObjectFromObject | ObjectSetFromObjects
ReferencedObjects ::=
 DefinedObject | ParameterizedObject |
 DefinedObjectSet | ParameterizedObjectSet
ValueFromObject ::= ReferencedObjects "." FieldName
ValueSetFromObjects ::= ReferencedObjects "." FieldName
TypeFromObject ::= ReferencedObjects "." FieldName
ObjectFromObject ::= ReferencedObjects "." FieldName
ObjectSetFromObjects ::= ReferencedObjects "." FieldName
InstanceOfType ::= INSTANCE OF DefinedObjectClass
InstanceOfValue ::= Value

SÉRIES DES RECOMMANDATIONS UIT-T

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