

# UIT-T

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

# X.691

**Amendement 2**  
(06/2006)

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

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

---

Technologies de l'information – Règles de codage  
ASN.1: spécification des règles de codage compact

**Amendement 2: Prise en compte des types de  
temps**

Recommandation UIT-T X.691 (2002) – Amendement 2

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

<b>RÉSEAUX PUBLICS DE DONNÉES</b>	
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
<b>INTERCONNEXION DES SYSTÈMES OUVERTS</b>	
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
<b>INTERFONCTIONNEMENT DES RÉSEAUX</b>	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.379
<b>SYSTÈMES DE MESSAGERIE</b>	<b>X.400–X.499</b>
<b>ANNUAIRE</b>	<b>X.500–X.599</b>
<b>RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES</b>	
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
<b>Notation de syntaxe abstraite numéro un (ASN.1)</b>	<b>X.680–X.699</b>
<b>GESTION OSI</b>	
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
<b>SÉCURITÉ</b>	<b>X.800–X.849</b>
<b>APPLICATIONS OSI</b>	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.889
Applications génériques de l'ASN.1	X.890–X.899
<b>TRAITEMENT RÉPARTI OUVERT</b>	<b>X.900–X.999</b>
<b>SÉCURITÉ DES TÉLÉCOMMUNICATIONS</b>	<b>X.1000–</b>

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

## Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact

### Amendement 2

#### Prise en compte des types de temps

#### Résumé

Le présent amendement spécifie les codages PER pour le type **TIME**, pour les types utiles de temps (**DATE**, **TIME-OF-DAY**, **DATE-TIME** et **DURATION**) et pour les types définis de temps spécifiés dans le module ASN.1 **DefinedTimeTypes** de la Rec. UIT-T X.680/Amd.3 | ISO/CEI 8824-1/Amd.3, Annexe A *bis*.

Des réglages de propriétés pour les valeurs abstraites des types utiles et définis de temps sont spécifiés dans la Rec. UIT-T X.680/Amd.3 | ISO/CEI 8824-1/Amd.3, Annexe A *bis*. Des réglages de propriétés pour les types additionnels de temps sont définis par un concepteur d'application en utilisant la notation de sous-type par réglage de propriété (qui est visible par les règles PER). Ces réglages de propriétés sont utilisés pour déterminer des codages PER optimaux. En particulier, une **DATE** sera normalement codée sur 15 bits, une heure-du-jour **TIME-OF-DAY** sur 17 bits, une date-heure **DATE-TIME** sur 32 bits (quatre octets) et une période **DURATION** sera fréquemment codée sur 16 bits ou moins (deux octets).

#### Source

L'Amendement 2 de la Recommandation UIT-T X.691 (2002) a été approuvé le 13 juin 2006 par la Commission d'études 17 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8. Un texte identique est publié comme Norme Internationale ISO/CEI 8825-2, Amendement 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.

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

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

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

## NOTE

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

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

## DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas des renseignements les plus récents, il est vivement recommandé aux développeurs de consulter la base de données des brevets du TSB sous <http://www.itu.int/ITU-T/ipr/>.

© UIT 2007

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

## TABLE DES MATIÈRES

	<i>Page</i>
1) Table des matières .....	1
2) Nouveau paragraphe 9.3.11 <i>bis</i> .....	1
3) Nouveau paragraphe 28 <i>bis</i> .....	1



**NORME INTERNATIONALE  
RECOMMANDATION UIT-T**

**Technologies de l'information – Règles de codage ASN.1:  
spécification des règles de codage compact**

**Amendement 2**

**Prise en compte des types de temps**

**1) Table des matières**

*Mettre à jour la Table des matières comme suit:*

28 bis	Codage du type temps, des types utiles de temps, des types définis de temps et des types additionnels de temps
28 bis.1	Général
28 bis.2	Codage des sous-types avec le réglage de propriété <b>"Basic=Date"</b>
28 bis.3	Codage des sous-types avec le réglage de propriété <b>"Basic=Time"</b>
28 bis.4	Codage des sous-types avec le réglage de propriété <b>"Basic=Date-Time"</b>
28 bis.5	Codage des sous-types avec les réglages de propriétés <b>"Basic=Interval Interval-type=SE"</b>
28 bis.6	Codage des sous-types avec les réglages de propriétés <b>"Basic=Interval Interval-type=D"</b>
28 bis.7	Codage des sous-types avec les réglages de propriétés <b>"Basic=Interval Interval-type=SD"</b> ou <b>"Basic=Interval Interval-type=DE"</b>
28 bis.8	Codage des sous-types avec les réglages de propriétés <b>"Basic=Rec-Interval Interval-type=SE"</b>
28 bis.9	Codage des sous-types avec les réglages de propriétés <b>"Basic=Rec-Interval Interval-type=D"</b>
28 bis.10	Codage des sous-types avec les réglages de propriétés <b>"Basic=Rec-Interval Interval-type=SD"</b> ou <b>"Basic=Rec-Interval Interval-type=DE"</b>
28 bis.11	Codage des sous-types avec des réglages mixtes de la propriété <b>Basic</b>

**2) Nouveau paragraphe 9.3.11 bis**

*Insérer le paragraphe 9.3.11 bis après le paragraphe 9.3.11:*

**9.3.11 bis** Les contraintes de réglage de propriété sur le type temps (ou sur les types utiles et définis de temps) qui ne sont pas extensibles après l'application des règles de la Rec. UIT-T X.680 | ISO/CEI 8824-1, 48.3 à 48.5, sont visibles par les règles PER. Les contraintes de réglage de propriété qui sont extensibles ne sont pas visibles par les règles PER.

**3) Nouveau paragraphe 28 bis**

*Insérer le paragraphe 28 bis avant le paragraphe 29:*

28 bis	<b>Codage du type temps, des types utiles de temps, des types définis de temps et des types additionnels de temps</b>
28 bis.1	<b>Général</b>

**28 bis.1.1** Le codage des types utiles de temps, des types définis de temps et des types additionnels de temps est déterminé par les réglages de propriétés des valeurs abstraites de ces types. Des réglages de propriétés pour les valeurs abstraites des types utiles de temps et pour les types définis de temps sont spécifiés dans la Rec. UIT-T X.680 | ISO/CEI 8824-1, § 34 *bis.4* et Annexe A *bis* respectivement. Les réglages de propriétés pour les valeurs abstraites des types additionnels de temps sont déterminés par les réglages de propriétés du type parent, restreints par toute contrainte visible par les PER qui s'applique (voir § 9.3.11 *bis*).

**28 bis.1.2** Si toutes les valeurs abstraites du type à coder ont l'un des réglages de propriétés listés dans l'une des lignes de la colonne 2 du Tableau 2, alors ce type est codé comme si le type avec ses contraintes (s'il y en a) avait été remplacé par le type spécifié dans la même ligne de la colonne 3 du Tableau 2. Sinon, il est codé comme spécifié au § 28 *bis.11*.

NOTE – Si une propriété de temps (par exemple **Midnight**) n'est pas listée dans le Tableau 2 pour une ligne particulière, il n'y a pas de contrainte sur son réglage.

**28 bis.1.3** Pour que les lignes 24 à 32 soient applicables, toutes les valeurs abstraites du type doivent avoir la même valeur de *n* dans **F<sub>n</sub>**.

**28 bis.1.4** Les types spécifiés en colonne 3 du Tableau 2 sont définis (en utilisant la notation ASN.1) dans les § 28 *bis.2* à 28 *bis.10*, et sont supposés être définis dans un environnement d'**AUTOMATIC TAGS**.

NOTE 1 – L'utilisation de ces noms de référence de type dans la spécification de codages PER ne les rend pas disponibles pour être utilisés par un concepteur d'application dans une spécification ASN.1; ils ne constituent pas non plus des mots réservés dans une telle spécification. Cependant, lorsque le suffixe **-ENCODING** est retiré, ils correspondent aux noms des types utiles ou définis de temps spécifiés dans la Rec. UIT-T X.680 | ISO/CEI 8824-1, § 34 *bis.4* et Annexe A *bis*.

NOTE 2 – Tous les types utiles et définis de temps satisfont les conditions pour l'une des lignes du Tableau 2 et ont par conséquent des codages optimisés. Des types additionnels de temps peuvent satisfaire les conditions pour l'une de ces lignes mais sont sinon codés comme spécifié au § 28 *bis.11*. Le type **TIME** non contraint est toujours codé comme spécifié au § 28 *bis.11*.

**Tableau 2 – Codage d'un sous-type de temps avec toutes les valeurs abstraites qui ont les réglages de propriétés spécifiés**

Numéro de la ligne	Réglages de propriétés	Type ASN.1 à coder
1	"Basic=Date Date=C Year=Basic" ou "Basic=Date Date=C Year=Proleptic"	<b>CENTURY-ENCODING</b> (voir § 28 <i>bis.2.1</i> )
2	"Basic=Date Date=C Year=Negative" ou "Basic=Date Date=C Year=Ln" (pour tout <i>n</i> )	<b>ANY-CENTURY-ENCODING</b> (voir § 28 <i>bis.2.2</i> )
3	"Basic=Date Date=Y Year=Basic" ou "Basic=Date Date=Y Year=Proleptic"	<b>YEAR-ENCODING</b> (voir § 28 <i>bis.2.3</i> )
4	"Basic=Date Date=Y Year=Negative" ou "Basic=Date Date=Y Year=Ln" (pour tout <i>n</i> )	<b>ANY-YEAR-ENCODING</b> (voir § 28 <i>bis.2.4</i> )
5	"Basic=Date Date=YM Year=Basic" ou "Basic=Date Date=YM Year=Proleptic"	<b>YEAR-MONTH-ENCODING</b> (voir § 28 <i>bis.2.5</i> )
6	"Basic=Date Date=YM Year=Negative" ou "Basic=Date Date=YM Year=Ln" (pour tout <i>n</i> )	<b>ANY-YEAR-MONTH-ENCODING</b> (voir § 28 <i>bis.2.6</i> )
7	"Basic=Date Date=YMD Year=Basic" ou "Basic=Date Date=YMD Year=Proleptic"	<b>DATE-ENCODING</b> (voir § 28 <i>bis.2.7</i> )



**Tableau 2 – Codage d'un sous-type de temps avec toutes les valeurs abstraites  
qui ont les réglages de propriétés spécifiés**

Numéro de la ligne	Réglages de propriétés	Type ASN.1 à coder
8	"Basic=Date Date=YMD Year=Negative" ou "Basic=Date Date=YMD Year=Ln" (pour tout <i>n</i> )	<b>ANY-DATE-ENCODING</b> (voir § 28 bis.2.8)
9	"Basic=Date Date=YD Year=Basic" ou "Basic=Date Date=YD Year=Proleptic"	<b>YEAR-DAY-ENCODING</b> (voir § 28 bis.2.9)
10	"Basic=Date Date=YD Year=Negative" ou "Basic=Date Date=YD Year=Ln" (pour tout <i>n</i> )	<b>ANY-YEAR-DAY-ENCODING</b> (voir § 28 bis.2.10)
11	"Basic=Date Date=YW Year=Basic" ou "Basic=Date Date=YW Year=Proleptic"	<b>YEAR-WEEK-ENCODING</b> (voir § 28 bis.2.11)
12	"Basic=Date Date=YW Year=Negative" ou "Basic=Date Date=YW Year=Ln" (pour tout <i>n</i> )	<b>ANY-YEAR-WEEK-ENCODING</b> (voir § 28 bis.2.12)
13	"Basic=Date Date=YWD Year=Basic" ou "Basic=Date Date=YWD Year=Proleptic"	<b>YEAR-WEEK-DAY-ENCODING</b> (voir § 28 bis.2.13)
14	"Basic=Date Date=YWD Year=Negative" ou "Basic=Date Date=YWD Year=Ln" (pour tout <i>n</i> )	<b>ANY-YEAR-WEEK-DAY-ENCODING</b> (voir § 28 bis.2.14)
15	"Basic=Time Time=H Local-or-UTC=L"	<b>HOURS-ENCODING</b> (voir § 28 bis.3.1)
16	"Basic=Time Time=H Local-or-UTC=Z"	<b>HOURS-UTC-ENCODING</b> (voir § 28 bis.3.2)
17	"Basic=Time Time=H Local-or-UTC=LD"	<b>HOURS-AND-DIFF-ENCODING</b> (voir § 28 bis.3.3)
18	"Basic=Time Time=HM Local-or-UTC=L"	<b>MINUTES-ENCODING</b> (voir § 28 bis.3.4)
19	"Basic=Time Time=HM Local-or-UTC=Z"	<b>MINUTES-UTC-ENCODING</b> (voir § 28 bis.3.5)
20	"Basic=Time Time=HM Local-or-UTC=LD"	<b>MINUTES-AND-DIFF-ENCODING</b> (voir § 28 bis.3.6)
21	"Basic=Time Time=HMS Local-or-UTC=L"	<b>TIME-OF-DAY-ENCODING</b> (voir § 28 bis.3.7)
22	"Basic=Time Time=HMS Local-or-UTC=Z"	<b>TIME-OF-DAY-UTC-ENCODING</b> (voir § 28 bis.3.8)
23	"Basic=Time Time=HMS Local-or-UTC=LD"	<b>TIME-OF-DAY-AND-DIFF-ENCODING</b> (voir § 28 bis.3.9)
24	"Basic=Time Time=HF <sub>n</sub> Local-or-UTC=L" (mais voir § 28 bis.1.3)	<b>HOURS-AND-FRACTION-ENCODING</b> (voir § 28 bis.3.10)
25	"Basic=Time Time=HF <sub>n</sub> Local-or-UTC=Z" (mais voir § 28 bis.1.3)	<b>HOURS-UTC-AND-FRACTION-ENCODING</b> (voir § 28 bis.3.11)

**Tableau 2 – Codage d'un sous-type de temps avec toutes les valeurs abstraites  
qui ont les réglages de propriétés spécifiés**

Numéro de la ligne	Réglages de propriétés	Type ASN.1 à coder
26	"Basic=Time Time=HFn Local-or-UTC=LD" (mais voir § 28 bis.1.3)	HOURS-AND-DIFF-AND-FRACTION-ENCODING (voir § 28 bis.3.12)
27	"Basic=Time Time=HMFn Local-or-UTC=L" (mais voir § 28 bis.1.3)	MINUTES-AND-FRACTION-ENCODING (voir § 28 bis.3.13)
28	"Basic=Time Time=HMFn Local-or-UTC=Z" (mais voir § 28 bis.1.3)	MINUTES-UTC-AND-FRACTION-ENCODING (voir § 28 bis.3.14)
29	"Basic=Time Time=HMFn Local-or-UTC=LD" (mais voir § 28 bis.1.3)	MINUTES-AND-DIFF-AND-FRACTION-ENCODING (voir § 28 bis.3.15)
30	"Basic=Time Time=HMSFn Local-or-UTC=L" (mais voir § 28 bis.1.3)	TIME-OF-DAY-AND-FRACTION-ENCODING (voir § 28 bis.3.16)
31	"Basic=Time Time=HMSFn Local-or-UTC=Z" (mais voir § 28 bis.1.3)	TIME-OF-DAY-UTC-AND-FRACTION-ENCODING (voir § 28 bis.3.17)
32	"Basic=Time Time=HMSFn Local-or-UTC=LD" (mais voir § 28 bis.1.3)	TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING (voir § 28 bis.3.18)
33	"Basic=Date-Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date" ainsi que les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	DATE-TIME-ENCODING {Date-Type, Time-Type} (instancié comme spécifié au § 28 bis.4.1)
34	"Basic=Interval Interval-type=SE SE-point=Date" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date".	START-END-DATE-INTERVAL-ENCODING {Date-Type} (voir § 28 bis.5.1)
35	"Basic=Interval Interval-type=SE SE-point=Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	START-END-TIME-INTERVAL-ENCODING {Time-Type} (voir § 28 bis.5.2)
36	"Basic=Interval Interval-type=SE SE-point=Date-Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date" ainsi que les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	START-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} (voir § 28 bis.5.3)
37	"Basic=Interval Interval-type=D"	DURATION-INTERVAL-ENCODING (voir § 28 bis.6.1)
38	"Basic=Interval Interval-type=SD SE-point=Date" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date".	START-DATE-DURATION-INTERVAL-ENCODING {Date-Type} (voir § 28 bis.7.1)
39	"Basic=Interval Interval-type=SD SE-point=Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	START-TIME-DURATION-INTERVAL-ENCODING {Time-Type} (voir § 28 bis.7.2)

**Tableau 2 – Codage d'un sous-type de temps avec toutes les valeurs abstraites qui ont les réglages de propriétés spécifiés**

Numéro de la ligne	Réglages de propriétés	Type ASN.1 à coder
40	"Basic=Interval Interval-type=SD SE-point=Date-Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date" ainsi que les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	START-DATE-TIME-DURATION-INTERVAL-ENCODING {Date-Type, Time-Type} (voir § 28 bis.7.3)
41	"Basic=Interval Interval-type=DE SE-point=Date" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date".	DURATION-END-DATE-INTERVAL-ENCODING {Date-Type} (voir § 28 bis.7.4)
42	"Basic=Interval Interval-type=DE SE-point=Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	DURATION-END-TIME-INTERVAL-ENCODING {Time-Type} (voir § 28 bis.7.5)
43	"Basic=Interval Interval-type=DE SE-point=Date-Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date" ainsi que les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	DURATION-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} (voir § 28 bis.7.6)
44	"Basic=Rec-Interval Interval-type=SE SE-point=Date" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date".	REC-START-END-DATE-INTERVAL-ENCODING {Date-Type} (voir § 28 bis.8.1)
45	"Basic=Rec-Interval Interval-type=SE SE-point=Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	REC-START-END-TIME-INTERVAL-ENCODING {Time-Type} (voir § 28 bis.8.2)
46	"Basic=Rec-Interval Interval-type=SE SE-point=Date-Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date" ainsi que les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	REC-START-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} (voir § 28 bis.8.3)
47	"Basic=Rec-Interval Interval-type=D" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	REC-DURATION-INTERVAL-ENCODING (voir § 28 bis.9.1)
48	"Basic=Rec-Interval Interval-type=SD SE-point=Date" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date".	REC-START-DATE-DURATION-INTERVAL-ENCODING {Date-Type} (voir § 28 bis.10.1)
49	"Basic=Rec-Interval Interval-type=SD SE-point=Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	REC-START-TIME-DURATION-INTERVAL-ENCODING {Time-Type} (voir § 28 bis.10.2)

**Tableau 2 – Codage d'un sous-type de temps avec toutes les valeurs abstraites qui ont les réglages de propriétés spécifiés**

Numéro de la ligne	Réglages de propriétés	Type ASN.1 à coder
50	"Basic=Rec-Interval Interval-type=SD SE-point=Date-Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date" ainsi que les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	REC-START-DATE-TIME-DURATION-INTERVAL-ENCODING  {Date-Type, Time-Type} (voir § 28 bis.10.3)
51	"Basic=Rec-Interval Interval-type=DE SE-point=Date" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date".	REC-DURATION-END-DATE-INTERVAL-ENCODING  {Date-Type} (voir § 28 bis.10.4)
52	"Basic=Rec-Interval Interval-type=DE SE-point=Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	REC-DURATION-END-TIME-INTERVAL-ENCODING  {Time-Type} (voir § 28 bis.10.5)
53	"Basic=Rec-Interval Interval-type=DE SE-point=Date-Time" Toutes les valeurs abstraites doivent avoir les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 1 à 14 pour "Basic=Date" ainsi que les mêmes réglages additionnels de propriétés spécifiés dans l'une des lignes 15 à 32 pour "Basic=Time".	REC-DURATION-END-DATE-TIME-INTERVAL-ENCODING  {Date-Type, Time-Type} (voir § 28 bis.10.6)

### 28 bis.2 Codage des sous-types avec le réglage de propriété "Basic=Date"

Ce paragraphe définit les types ASN.1 référencés dans le Tableau 2, colonne 3 pour des types dont toutes les valeurs abstraites ont le réglage de propriété "Basic=Date".

**28 bis.2.1** Le type CENTURY-ENCODING est défini comme:

CENTURY-ENCODING ::= INTEGER(0..99) -- 7 bits

avec la valeur entière qui prend la valeur spécifiée par les deux premiers chiffres du composant année de la valeur abstraite.

**28 bis.2.2** Le type ANY-CENTURY-ENCODING est défini comme:

ANY-CENTURY-ENCODING ::= INTEGER(MIN..MAX)

avec la valeur entière qui prend la valeur spécifiée par le composant année de la valeur abstraite, en ignorant les deux derniers chiffres.

**28 bis.2.3** Le type YEAR-ENCODING est défini comme:

YEAR-ENCODING ::= CHOICE { -- 2 bits pour le déterminant de choix  
     immediate       INTEGER (2005..2020), -- 4 bits  
     near-future     INTEGER (2021..2276), -- 8 bits  
     near-past       INTEGER (1749..2004), -- 8 bits  
     remainder      INTEGER (MIN..1748 | 2277..MAX) }

avec la valeur entière qui prend la valeur du composant année de la valeur abstraite.

NOTE – Ceci a été optimisé pour fournir un codage sur 6 ou 10 bits pour des cas communs.

**28 bis.2.4** Le type ANY-YEAR-ENCODING est défini comme:

ANY-YEAR-ENCODING ::= INTEGER(MIN..MAX)

avec la valeur entière qui prend la valeur du composant année de la valeur abstraite.

**28 bis.2.5** Le type **YEAR-MONTH-ENCODING** est défini comme:

```
YEAR-MONTH-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    month        INTEGER (1..12) -- 4 bits -- }
```

avec la valeur du composant **YEAR-ENCODING** fixée en accord avec le § 28 bis.2.3 et le composant entier **month** qui prend la valeur du composant mois de la valeur abstraite.

NOTE – Ceci a été optimisé pour fournir un codage sur 10 ou 14 bits pour des cas communs.

**28 bis.2.6** Le type **ANY-YEAR-MONTH-ENCODING** est défini comme:

```
ANY-YEAR-MONTH-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    month        INTEGER (1..12) }
```

avec la valeur du composant **ANY-YEAR-ENCODING** fixée en accord avec le § 28 bis.2.4 et le composant entier **month** qui prend la valeur du composant mois de la valeur abstraite.

**28 bis.2.7** Le type **DATE-ENCODING** est défini comme:

```
DATE-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    month        INTEGER (1..12), -- 4 bits
    day          INTEGER (1..31) -- 5 bits -- }
```

avec la valeur du composant **YEAR-ENCODING** fixée en accord avec le § 28 bis.2.3, le composant entier **month** qui prend la valeur du composant mois de la valeur abstraite et le composant entier **day** qui prend la valeur du composant jour de la valeur abstraite.

NOTE – Ceci a été optimisé pour fournir un codage sur 15 ou 19 bits pour des cas communs.

**28 bis.2.8** Le type **ANY-DATE-ENCODING** est défini comme:

```
ANY-DATE-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    month        INTEGER (1..12),
    day          INTEGER (1..31) }
```

avec la valeur du composant **ANY-YEAR-ENCODING** fixée en accord avec le § 28 bis.2.4, le composant entier **month** qui prend la valeur du composant mois de la valeur abstraite et le composant entier **day** qui prend la valeur du composant jour de la valeur abstraite.

**28 bis.2.9** Le type **YEAR-DAY-ENCODING** est défini comme:

```
YEAR-DAY-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    day          INTEGER (1..366) }
```

avec la valeur du composant **YEAR-ENCODING** fixée en accord avec le § 28 bis.2.3 et le composant entier **day** qui prend la valeur du composant jour de la valeur abstraite.

**28 bis.2.10** Le type **ANY-YEAR-DAY-ENCODING** est défini comme:

```
ANY-YEAR-DAY-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    day          INTEGER (1..366) }
```

avec la valeur du composant **ANY-YEAR-ENCODING** fixée en accord avec le § 28 bis.2.4 et le composant entier **day** qui prend la valeur du composant jour de la valeur abstraite.

**28 bis.2.11** Le type **YEAR-WEEK-ENCODING** est défini comme:

```
YEAR-WEEK-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    week         INTEGER (1..53), -- 6 bits -- }
```

avec la valeur du composant **YEAR-ENCODING** fixée en accord avec le § 28 bis.2.3 et le composant entier **week** qui prend la valeur du composant semaine de la valeur abstraite.

NOTE – Ceci a été optimisé pour fournir un codage sur 12 ou 16 bits pour des cas communs.

**28 bis.2.12** Le type **ANY-YEAR-WEEK-ENCODING** est défini comme:

```
ANY-YEAR-WEEK-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    week          INTEGER (1..53)}
```

avec la valeur du composant **ANY-YEAR-ENCODING** fixée en accord avec le § 28 bis.2.4 et le composant entier **week** qui prend la valeur du composant semaine de la valeur abstraite.

**28 bis.2.13** Le type **YEAR-WEEK-DAY-ENCODING** est défini comme:

```
YEAR-WEEK-DAY-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    week          INTEGER (1..53), -- 6 bits
    day           INTEGER (1..7) -- 3 bits -- }
```

avec la valeur du composant **YEAR-ENCODING** fixée en accord avec le § 28 bis.2.3, le composant entier **week** qui prend la valeur du composant semaine de la valeur abstraite et le composant entier **day** qui prend la valeur du composant jour de la valeur abstraite.

NOTE – Ceci a été optimisé pour fournir un codage sur 15 ou 19 bits pour des cas communs.

**28 bis.2.14** Le type **ANY-YEAR-WEEK-DAY-ENCODING** est défini comme:

```
ANY-YEAR-WEEK-DAY-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    week          INTEGER (1..53),
    day           INTEGER (1..7)}
```

avec la valeur du composant **ANY-YEAR-ENCODING** fixée en accord avec le § 28 bis.2.4, le composant entier **week** qui prend la valeur du composant semaine de la valeur abstraite et le composant entier **day** qui prend la valeur du composant jour de la valeur abstraite.

### **28 bis.3 Codage des sous-types avec le réglage de propriété "Basic=Time"**

Ce paragraphe définit les types ASN.1 référencés dans le Tableau 2, colonne 3 pour des types dont toutes les valeurs abstraites ont le réglage de propriété **Basic=Time**.

**28 bis.3.1** Le type **HOURS-ENCODING** est défini comme:

```
HOURS-ENCODING ::= INTEGER(0..24) -- 5 bits
```

avec la valeur entière qui prend la valeur du composant heures de la valeur abstraite.

NOTE – Ceci a été optimisé pour fournir un codage sur 5 bits.

**28 bis.3.2** Le type **HOURS-UTC-ENCODING** est défini comme:

```
HOURS-UTC-ENCODING ::= INTEGER(0..24) -- 5 bits
```

avec la valeur entière qui prend la valeur du composant heures de la valeur abstraite.

NOTE – Ceci a été optimisé pour fournir un codage sur 5 bits.

**28 bis.3.3** Le type **HOURS-AND-DIFF-ENCODING** est défini comme:

```
HOURS-AND-DIFF-ENCODING ::= SEQUENCE {
    local-hours   INTEGER (0..24),
    time-difference TIME-DIFFERENCE }
```

où:

```
TIME-DIFFERENCE ::= SEQUENCE {
    hours         INTEGER (-15..16),
    minutes       INTEGER (1..59) OPTIONAL }
```

avec le composant entier **local-hours** qui prend la valeur du composant heures de la valeur abstraite et le composant **time-difference** qui prend les valeurs des heures et minutes du composant différence d'heure de la valeur abstraite. Si le composant minutes de la différence d'heure est nul, le composant **minutes** du type **TIME-DIFFERENCE** est omis.

**28 bis.3.4** Le type **MINUTES-ENCODING** est défini comme:

```
MINUTES-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59) -- 5 bits -- }
```

avec le composant entier **hours** qui prend la valeur du composant heures de la valeur abstraite et le composant entier **minutes** qui prend la valeur du composant minutes de la valeur abstraite.

NOTE – Ceci a été optimisé pour fournir un codage sur 10 bits.

**28 bis.3.5** Le type **MINUTES-UTC-ENCODING** est défini comme:

```
MINUTES-UTC-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59) -- 5 bits -- }
```

avec le composant entier **hours** qui prend la valeur du composant heures de la valeur abstraite et le composant entier **minutes** qui prend la valeur du composant minutes.

NOTE – Ceci a été optimisé pour fournir un codage sur 10 bits.

**28 bis.3.6** Le type **MINUTES-AND-DIFF-ENCODING** est défini comme:

```
MINUTES-AND-DIFF-ENCODING ::= SEQUENCE {
    local-time     SEQUENCE {
        hours      INTEGER (0..24),
        minutes    INTEGER (0..59) },
    time-difference TIME-DIFFERENCE }
```

avec le composant **local-time** qui prend la valeur des composants heures et minutes de l'heure locale de la valeur abstraite et le composant **time-difference** qui prend les valeurs des heures et minutes du composant différence d'heure de la valeur abstraite comme spécifié au § 28 bis.3.3.

**28 bis.3.7** Le type **TIME-OF-DAY-ENCODING** est défini comme:

```
TIME-OF-DAY-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    seconds        INTEGER (0..60) -- 5 bits -- }
```

avec le composant entier **hours** qui prend la valeur du composant heures de la valeur abstraite, le composant entier **minutes** qui prend la valeur du composant minutes et le composant entier **seconds** qui prend la valeur du composant secondes.

NOTE – Ceci a été optimisé pour fournir un codage sur 15 bits.

**28 bis.3.8** Le type **TIME-OF-DAY-UTC-ENCODING** est défini comme:

```
TIME-OF-DAY-UTC-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    seconds        INTEGER (0..60) -- 5 bits -- }
```

avec le composant entier **hours** qui prend la valeur du composant heures de la valeur abstraite, le composant entier **minutes** qui prend la valeur du composant minutes et le composant entier **seconds** qui prend la valeur du composant secondes.

NOTE – Ceci a été optimisé pour fournir un codage sur 15 bits.

**28 bis.3.9** Le type **TIME-OF-DAY-AND-DIFF-ENCODING** est défini comme:

```
TIME-OF-DAY-AND-DIFF-ENCODING ::= SEQUENCE {
    local-time     SEQUENCE {
        hours      INTEGER (0..24),
        minutes    INTEGER (0..59),
        seconds    INTEGER (0..60) },
    time-difference TIME-DIFFERENCE }
```

avec le composant **local-time** qui prend la valeur des composants heures, minutes et secondes de l'heure locale de la valeur abstraite et le composant **time-difference** qui prend les valeurs des heures et minutes du composant différence d'heure de la valeur abstraite comme spécifié au § 28 bis.3.3.

**28 bis.3.10** Le type **HOURS-AND-FRACTION-ENCODING** est défini comme:

```
HOURS-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    fraction       INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits pour une précision jusqu'à trois chiffres -- }
```

avec le composant entier **hours** qui prend la valeur du composant heures de la valeur abstraite et le composant entier **fraction** qui prend la valeur des heures décimales multipliée par 10 à la puissance N, où N est le nombre spécifié de chiffres dans la partie décimale.

NOTE – Ceci a été optimisé pour fournir un codage sur 16 bits jusqu'à une précision de 3 chiffres.

**28 bis.3.11** Le type **HOURS-UTC-AND-FRACTION-ENCODING** est défini comme:

```
HOURS-UTC-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    fraction       INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits pour une précision jusqu'à trois chiffres -- }
```

avec le composant entier **hours** qui prend la valeur du composant heures de la valeur abstraite et le composant entier **fraction** qui prend la valeur des heures décimales multipliée par 10 à la puissance N, où N est le nombre spécifié de chiffres dans la partie décimale.

NOTE – Ceci a été optimisé pour fournir un codage sur 16 bits jusqu'à une précision de 3 chiffres.

**28 bis.3.12** Le type **HOURS-AND-DIFF-AND-FRACTION-ENCODING** est défini comme:

```
HOURS-AND-DIFF-AND-FRACTION-ENCODING ::= SEQUENCE {
    local-hours    INTEGER (0..24), -- 5 bits
    fraction       INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits pour une précision jusqu'à trois chiffres -- ,
    time-difference TIME-DIFFERENCE }
```

avec le composant **local-hours** qui prend la valeur du composant heures de l'heure locale de la valeur abstraite, le composant entier **fraction** qui prend la valeur des heures décimales multipliée par 10 à la puissance N (où N est le nombre spécifié de chiffres dans la partie décimale) et le composant **time-difference** qui prend les valeurs des heures et minutes du composant différence d'heure de la valeur abstraite comme spécifié au § 28 bis.3.3.

**28 bis.3.13** Le type **MINUTES-AND-FRACTION-ENCODING** est défini comme:

```
MINUTES-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    fraction       INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits pour une précision jusqu'à trois chiffres -- }
```

avec le composant entier **hours** qui prend la valeur du composant heures de la valeur abstraite, le composant entier **minutes** qui prend la valeur du composant minutes et le composant entier **fraction** qui prend la valeur des heures décimales multipliée par 10 à la puissance N, où N est le nombre spécifié de chiffres dans la partie décimale.

NOTE – Ceci a été optimisé pour fournir un codage sur 21 bits jusqu'à une précision de 3 chiffres.

**28 bis.3.14** Le type **MINUTES-UTC-AND-FRACTION-ENCODING** est défini comme:

```
MINUTES-UTC-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    fraction       INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits pour une précision jusqu'à trois chiffres -- }
```

avec le composant entier **hours** qui prend la valeur du composant heures de la valeur abstraite, le composant entier **minutes** qui prend la valeur du composant minutes et le composant entier **fraction** qui prend la valeur des heures décimales multipliée par 10 à la puissance N, où N est le nombre spécifié de chiffres dans la partie décimale.

NOTE – Ceci a été optimisé pour fournir un codage sur 21 bits jusqu'à une précision de 3 chiffres.

**28 bis.3.15** Le type **MINUTES-AND-DIFF-AND-FRACTION-ENCODING** est défini comme:

```
MINUTES-AND-DIFF-AND-FRACTION-ENCODING ::= SEQUENCE {
    local-time     SEQUENCE {
        hours      INTEGER (0..24),
        minutes    INTEGER (0..59),
        fraction    INTEGER (0..999, ..., 1000..MAX)},
    time-difference TIME-DIFFERENCE }
```



avec le composant `local-time` qui prend la valeur des composants heures et minutes de l'heure locale de la valeur abstraite, le composant entier `fraction` qui prend la valeur des heures décimales multipliée par 10 à la puissance N (où N est le nombre spécifié de chiffres dans la partie décimale) et le composant `time-difference` qui prend les valeurs des heures et minutes du composant différence d'heure de la valeur abstraite comme spécifié au § 28 bis.3.3.

**28 bis.3.16** Le type `TIME-OF-DAY-AND-FRACTION-ENCODING` est défini comme:

```
TIME-OF-DAY-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    seconds        INTEGER (0..60), -- 5 bits --
    fraction       INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits pour une précision jusqu'à trois chiffres -- }
```

avec le composant entier `hours` qui prend la valeur du composant heures de la valeur abstraite, le composant entier `minutes` qui prend la valeur du composant minutes, le composant entier `seconds` qui prend la valeur du composant secondes et le composant entier `fraction` qui prend la valeur des heures décimales multipliée par 10 à la puissance N, où N est le nombre spécifié de chiffres dans la partie décimale.

NOTE – Ceci a été optimisé pour fournir un codage sur 26 bits.

**28 bis.3.17** Le type `TIME-OF-DAY-UTC-AND-FRACTION-ENCODING` est défini comme:

```
TIME-OF-DAY-UTC-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    seconds        INTEGER (0..60), -- 5 bits --
    fraction       INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits pour une précision jusqu'à trois chiffres -- }
```

avec le composant entier `hours` qui prend la valeur du composant heures de la valeur abstraite, le composant entier `minutes` qui prend la valeur du composant minutes de la valeur abstraite, le composant entier `seconds` qui prend la valeur du composant secondes de la valeur abstraite et le composant entier `fraction` qui prend la valeur des heures décimales multipliée par 10 à la puissance N, où N est le nombre spécifié de chiffres dans la partie décimale.

NOTE – Ceci a été optimisé pour fournir un codage sur 26 bits.

**28 bis.3.18** Le type `TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING` est défini comme:

```
TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING ::= SEQUENCE {
    local-time     SEQUENCE {
        hours      INTEGER (0..24),
        minutes    INTEGER (0..59),
        seconds    INTEGER (0..60),
        fraction    INTEGER (0..999, ..., 1000..MAX)},
    time-difference TIME-DIFFERENCE }
```

avec le composant `local-time` qui prend la valeur des composants heures, minutes, secondes et partie décimale de l'heure locale de la valeur abstraite et le composant `time-difference` qui prend les valeurs des heures et minutes du composant différence d'heure de la valeur abstraite comme spécifié au § 28 bis.3.3.

**28 bis.4** **Codage des sous-types avec le réglage de propriété "Basic=Date-Time"**

Le présent paragraphe définit le type ASN.1 référencé dans le Tableau 2, colonne 3 pour des types dont toutes les valeurs abstraites ont le réglage de propriété "Basic=Date-Time".

**28 bis.4.1** Le type `DATE-TIME-ENCODING` est défini comme:

```
DATE-TIME-ENCODING {Date-Type, Time-Type} ::= SEQUENCE {
    date      Date-Type,
    time      Time-Type}
```

**28 bis.4.2** Le codage est celui d'une instanciation de ce type avec les paramètres réels `Date-Type` et `Time-Type` qui valent les types spécifiés dans le Tableau 2, colonne 3 des lignes "Basic=Date" et "Basic=Time" (respectivement) qui spécifient les réglages additionnels de propriétés de toutes les valeurs abstraites de ce type.

NOTE – Ceci a été optimisé pour fournir un codage sur 32 bits pour des cas communs.

**28 bis.5** **Codage des sous-types avec les réglages de propriétés "Basic=Interval Interval-type=SE"**

Le présent paragraphe définit les types ASN.1 référencés dans le Tableau 2, colonne 3 pour des types dont toutes les valeurs abstraites ont le réglage de propriété "Basic=Interval Interval-type=SE".

**28 bis.5.1** Le type **START-END-DATE-INTERVAL-ENCODING** est défini comme:

```
START-END-DATE-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    start          Date-Type,
    end            Date-Type}
```

et le codage est celui d'une instantiation de ce type avec le paramètre réel **Date-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "**Basic=Date**" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **start** prend la valeur de la date de début et le composant **end** prend la valeur de la date de fin de l'intervalle.

**28 bis.5.2** Le type **START-END-TIME-INTERVAL-ENCODING** est défini comme:

```
START-END-TIME-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    start          Time-Type,
    end            Time-Type}
```

et le codage est celui d'une instantiation de ce type avec le paramètre réel **Time-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "**Basic=Time**" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **start** prend la valeur de la date de début et le composant **end** prend la valeur de la date de fin de l'intervalle.

**28 bis.5.3** Le type **START-END-DATE-TIME-INTERVAL-ENCODING** est défini comme:

```
START-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
SEQUENCE {
    start          DATE-TIME-ENCODING {Date-Type, Time-Type},
    end            DATE-TIME-ENCODING {Date-Type, Time-Type}}
```

et le codage est celui d'une instantiation de ce type avec les paramètres réels **Date-Type** et **Time-Type** qui valent le type spécifié dans le Tableau 2, colonne 3 des lignes "**Basic=Date**" et "**Basic=Time**" (respectivement) qui spécifient les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **start** prend la valeur (comme spécifié au § 28 bis.4) de la date de début et le composant **end** prend la valeur de la date de fin de l'intervalle.

**28 bis.6** **Codage des sous-types avec les réglages de propriétés "Basic=Interval Interval-type=D"**

Le présent paragraphe définit le type ASN.1 référencé dans le Tableau 2, colonne 3 pour des types dont toutes les valeurs abstraites ont le réglage de propriété "**Basic=Interval Interval-type=D**".

**28 bis.6.1** Le type **DURATION-INTERVAL-ENCODING** est défini comme:

```
DURATION-INTERVAL-ENCODING ::= SEQUENCE { -- 8 bits pour l'optionalité
    years          INTEGER (0..31, ..., 32..MAX) OPTIONAL,
                  -- 5 bits pour jusqu'à 31 années
    months         INTEGER (0..15, ..., 16..MAX) OPTIONAL,
                  -- 4 bits pour jusqu'à 15 mois
    weeks          INTEGER (0..63, ..., 64..MAX) OPTIONAL,
                  -- 6 bits pour jusqu'à 63 semaines
    days           INTEGER (0..31, ..., 32..MAX) OPTIONAL,
                  -- 5 bits pour jusqu'à 31 jours
    hours          INTEGER (0..31, ..., 32..MAX) OPTIONAL,
                  -- 5 bits pour jusqu'à 31 heures
    minutes        INTEGER (0..63, ..., 64..MAX) OPTIONAL,
                  -- 6 bits pour jusqu'à 63 minutes
    seconds        INTEGER (0..63, ..., 64..MAX) OPTIONAL,
                  -- 6 bits pour jusqu'à 63 secondes
    fractional-part SEQUENCE {
        number-of-digits  INTEGER(1..3, ..., 4..MAX),
                          -- 3 bits pour jusqu'à trois chiffres de précision
        fractional-value   INTEGER(1..999, ..., 1000..MAX)
                          -- 11 bits pour jusqu'à trois chiffres de précision
    } OPTIONAL }
```

**28 bis.6.2** Le composant **weeks** est présent si et seulement si les composants **years**, **months**, **days**, **hours**, **minutes** et **seconds** sont tous absents.

NOTE – Ceci reflète les restrictions qui sont présentes pour l'utilisation d'éléments de temps dans la définition de la valeur abstraite **DURATION**.

**28 bis.6.3** (Mise en forme canonique) Si un composant élément de temps de la valeur abstraite est nul et n'a pas de partie décimale, alors le composant correspondant du type **DURATION-INTERVAL-ENCODING** est absent à moins que cet élément de temps soit le moins significatif dans la valeur abstraite. Si un élément de temps de la valeur abstraite est nul

et est l'élément de temps le moins significatif dans la valeur abstraite ou a une partie décimale, alors le composant correspondant doit être présent avec une valeur nulle dans le type **DURATION-INTERVAL-ENCODING**.

**28 bis.6.4** Le composant **fractional-part** du type **DURATION-INTERVAL-ENCODING** est absent si aucun élément de temps n'a de partie décimale, sinon il prend comme valeur la partie décimale (de l'élément de temps le moins significatif) comme spécifié au § 28 bis.6.5.

**28 bis.6.5** Le nombre de chiffres dans la partie décimale est placé dans le composant **number-of-digits**. Si le nombre de chiffres est N, alors la valeur de la partie décimale est multipliée par 10 à la puissance N et l'entier résultant est placé dans la composant **fractional-value**.

NOTE 1 – Les décodeurs peut retrouver la partie décimale d'origine depuis ces codages, y compris les éventuels zéros de fin.

NOTE 2 – Ce codage a été optimisé pour les cas où il n'y a que quelques éléments de temps non nuls dans la valeur abstraite et où les valeurs des éléments de temps sont petites. Des codages de moins de 16 bits se rencontrent dans les cas simples.

**28 bis.7** **Codage des sous-types avec les réglages de propriétés "Basic=Interval Interval-type=SD" ou "Basic=Interval Interval-type=DE"**

Le présent paragraphe définit les types ASN.1 références dans le Table 2, colonne 3 pour des types dont toutes les valeurs abstraites ont le réglage de propriété **"Basic=Interval Interval-type=SD"** ou **"Basic=Interval Interval-type=DE"**.

**28 bis.7.1** Le type **START-DATE-DURATION-INTERVAL-ENCODING** est défini comme:

```
START-DATE-DURATION-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    start          Date-Type,
    duration       DURATION-INTERVAL-ENCODING }
```

et le codage est celui d'une instantiation de ce type avec le paramètre réel **Date-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne **"Basic=Date"** qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **start** prend la valeur de la date de début et le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle.

**28 bis.7.2** Le type **START-TIME-DURATION-INTERVAL-ENCODING** est défini comme:

```
START-TIME-DURATION-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    start          Time-Type,
    duration       DURATION-INTERVAL-ENCODING }
```

et le codage est celui d'une instantiation de ce type avec le paramètre réel **Time-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne **"Basic=Time"** qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **start** prend la valeur de l'heure de début et le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle.

**28 bis.7.3** Le type **START-DATE-TIME-DURATION-INTERVAL-ENCODING** est défini comme:

```
START-DATE-TIME-DURATION-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
    SEQUENCE {
        start          DATE-TIME-ENCODING {Date-Type, Time-Type},
        duration       DURATION-INTERVAL-ENCODING }
```

et le codage est celui d'une instantiation de ce type avec les paramètres réels **Date-Type** et **Time-Type** qui valent les types spécifiés dans le Tableau 2, colonne 3 des lignes **"Basic=Date"** et **"Basic=Time"** (respectivement) qui spécifient les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **start** prend la valeur (comme spécifié au § 28 bis.4) de la date-heure de début et le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle.

**28 bis.7.4** Le type **DURATION-END-DATE-INTERVAL-ENCODING** est défini comme:

```
DURATION-END-DATE-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    duration       DURATION-INTERVAL-ENCODING,
    end           Date-Type }
```

et le codage est celui d'une instantiation de ce type avec le paramètre réel **Date-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne **"Basic=Date"** qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle et le composant **end** prend la valeur de la date de fin.

**28 bis.7.5** Le type **DURATION-END-TIME-INTERVAL-ENCODING** est défini comme:

```
DURATION-END-TIME-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    duration      DURATION-INTERVAL-ENCODING,
    end           Time-Type }
```

et le codage est celui d'une instanciation de ce type avec le paramètre réel **Time-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "**Basic=Time**" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle et le composant **end** prend la valeur de l'heure de fin.

**28 bis.7.6** Le type **DURATION-END-DATE-TIME-INTERVAL-ENCODING** est défini comme:

```
DURATION-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
SEQUENCE {
    duration      DURATION-INTERVAL-ENCODING,
    end           DATE-TIME-ENCODING {Date-Type, Time-Type}}
```

et le codage est celui d'une instanciation de ce type avec les paramètres réels **Date-Type** et **Time-Type** qui valent les types spécifiés dans le Tableau 2, colonne 3 des lignes "**Basic=Date**" et "**Basic=Time**" (respectivement) qui spécifient les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle et le composant **end** prend la valeur (comme spécifié au § 28 bis.4) de l'heure de fin.

**28 bis.8** **Codage des sous-types avec les réglages de propriétés "Basic=Rec-Interval Interval-type=SE"**

Le présent paragraphe définit les types ASN.1 référencés dans le Tableau 2, colonne 3 pour des types dont toutes les valeurs abstraites ont les réglages de propriétés "**Basic=Rec-Interval Interval-type=SE**".

**28 bis.8.1** Le type **REC-START-END-DATE-INTERVAL-ENCODING** est défini comme:

```
REC-START-END-DATE-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    recurrence    INTEGER OPTIONAL,
    start         Date-Type,
    end           Date-Type}
```

et le codage est celui d'une instanciation de ce type avec le paramètre réel **Date-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "**Basic=Date**" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **start** prend la valeur de la date de début et le composant **end** prend la valeur de la date de fin de l'intervalle.

**28 bis.8.2** Le type **REC-START-END-TIME-INTERVAL-ENCODING** est défini comme:

```
REC-START-END-TIME-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    recurrence    INTEGER OPTIONAL,
    start         Time-Type,
    end           Time-Type}
```

et le codage est celui d'une instanciation de ce type avec le paramètre réel **Time-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "**Basic=Time**" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **start** prend la valeur de l'heure de début et le composant **end** prend la valeur de l'heure de fin de l'intervalle.

**28 bis.8.3** Le type **REC-START-END-DATE-TIME-INTERVAL-ENCODING** est défini comme:

```
REC-START-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} ::= SEQUENCE {
    recurrence    INTEGER OPTIONAL,
    start         DATE-TIME-ENCODING {Date-Type, Time-Type},
    end           DATE-TIME-ENCODING {Date-Type, Time-Type}}
```

et le codage est celui d'une instanciation de ce type avec les paramètres réels **Date-Type** et **Time-Type** qui valent les types spécifiés dans le Tableau 2, colonne 3 des lignes "**Basic=Date**" et "**Basic=Time**" (respectivement) qui spécifient les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **start** prend la valeur (comme spécifié au § 28 bis.4) de la date-heure de début et le composant **end** prend la valeur de la date-heure de fin de l'intervalle.

**28 bis.9 Codage des sous-types avec les réglages de propriétés "Basic=Rec-Interval Interval-type=D"**

Le présent paragraphe définit le type ASN.1 référence dans le Tableau 2, colonne 3 pour des types dont toutes les valeurs abstraites ont les réglages de propriétés "Basic=Rec-Interval Interval-type=D".

**28 bis.9.1** Le type REC-DURATION-INTERVAL-ENCODING est défini comme:

```
REC-DURATION-INTERVAL-ENCODING ::= SEQUENCE {
    recurrence    INTEGER OPTIONAL,
    duration      DURATION-INTERVAL-ENCODING}
```

**28 bis.9.2** Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle répétitif.

**28 bis.10 Codage des sous-types avec les réglages de propriétés "Basic=Rec-Interval Interval-type=SD" ou "Basic=Rec-Interval Interval-type=DE"**

Le présent paragraphe définit les types ASN.1 référencés dans le Tableau 2, colonne 3 pour les types dont toutes les valeurs abstraites ont les réglages de propriété "Basic=Rec-Interval Interval-type=SD" ou "Basic=Rec-Interval Interval-type=DE".

**28 bis.10.1** Le type REC-START-DATE-DURATION-INTERVAL-ENCODING est défini comme:

```
REC-START-DATE-DURATION-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    recurrence    INTEGER OPTIONAL,
    start         Date-Type,
    duration      DURATION-INTERVAL-ENCODING}
```

et le codage est celui d'une instanciation de ce type avec le paramètre réel **Date-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "Basic=Date" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **start** prend la valeur de la date de début et le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période fin de l'intervalle.

**28 bis.10.2** Le type REC-START-TIME-DURATION-INTERVAL-ENCODING est défini comme:

```
REC-START-TIME-DURATION-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    recurrence    INTEGER OPTIONAL,
    start         Time-Type,
    duration      DURATION-INTERVAL-ENCODING }
```

et le codage est celui d'une instanciation de ce type avec le paramètre réel **Time-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "Basic=Time" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **start** prend la valeur de l'heure de début et le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période fin de l'intervalle.

**28 bis.10.3** Le type REC-START-DATE-TIME-DURATION-INTERVAL-ENCODING est défini comme:

```
REC-START-DATE-TIME-DURATION-INTERVAL-ENCODING {Date-Type, Time-Type}
 ::= SEQUENCE {
    recurrence    INTEGER OPTIONAL,
    start         DATE-TIME-ENCODING {Date-Type, Time-Type},
    duration      DURATION-INTERVAL-ENCODING }
```

et le codage est celui d'une instanciation de ce type avec les paramètres réels **Date-Type** et **Time-Type** qui valent les types spécifiés dans le Tableau 2, colonne 3 des lignes "Basic=Date" et "Basic=Time" (respectivement) qui spécifient les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **start** prend la valeur (comme spécifié au § 28 bis.4) de la date-heure de début et le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle répétitif.

**28 bis.10.4** Le type REC-DURATION-END-DATE-INTERVAL-ENCODING est défini comme:

```
REC-DURATION-END-DATE-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    recurrence    INTEGER OPTIONAL,
    duration      DURATION-INTERVAL-ENCODING,
    end           Date-Type }
```

et le codage est celui d'une instanciation de ce type avec le paramètre réel **Date-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "**Basic=Date**" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle et le composant **end** prend la valeur de la date de fin.

**28 bis.10.5** Le type **REC-DURATION-END-TIME-INTERVAL-ENCODING** est défini comme:

```
REC-DURATION-END-TIME-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    recurrence      INTEGER OPTIONAL,
    duration        DURATION-INTERVAL-ENCODING,
    end             Time-Type }
```

et le codage est celui d'une instanciation de ce type avec le paramètre réel **Time-Type** qui vaut le type spécifié dans le Tableau 2, colonne 3 de la ligne "**Basic=Time**" qui spécifie les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la période de l'intervalle et le composant **end** prend la valeur de l'heure de fin.

**28 bis.10.6** Le type **REC-DURATION-END-DATE-TIME-INTERVAL-ENCODING** est défini comme:

```
REC-DURATION-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
    SEQUENCE {
        recurrence      INTEGER OPTIONAL,
        duration        DURATION-INTERVAL-ENCODING,
        end             DATE-TIME-ENCODING {Date-Type, Time-Type}}
```

et le codage est celui d'une instanciation de ce type avec les paramètres réels **Date-Type** et **Time-Type** qui valent les types spécifiés dans le Tableau 2, colonne 3 des lignes "**Basic=Date**" et "**Basic=Time**" (respectivement) qui spécifient les réglages additionnels de propriétés de toutes les valeurs abstraites du type. Le composant **recurrence** est absent pour un nombre illimité de répétitions dans la valeur abstraite et prend autrement la valeur du nombre de répétitions. Le composant **duration** prend la valeur (comme spécifié au § 28 bis.6) de la durée de l'intervalle et le composant **end** prend la valeur (comme spécifié au § 28 bis.4) de la date de fin.

### **28 bis.11 Codage des sous-types avec des réglages mixtes de la propriété Basic**

Le présent paragraphe spécifie le codage du type **TIME** et des sous-ensembles de ce type dont les valeurs abstraites n'ont pas le même réglage de la propriété **Basic** ou auxquelles aucune ligne applicable du Tableau 2 ne peut être appliquée (par exemple, à cause de l'utilisation de différentes précisions - voir § 28 bis.1.3). Il définit et utilise les types **DATE-TYPE**, **TIME-TYPE** et **MIXED-ENCODING** (voir § 28 bis.11.5 et 28 bis.11.6). Ces types sont définis en utilisant les types ASN.1 définis dans des paragraphes précédents.

**28 bis.11.1** Pour chaque valeur abstraite du type **TIME**, il y a exactement une ligne du Tableau 2 pour laquelle les réglages de propriétés spécifiés en colonne 2 correspondent aux réglages de propriétés de la valeur abstraite. (La valeur abstraite peut avoir des réglages additionnels de propriétés non listés dans la colonne 2.) Cette ligne est appelée la ligne principale de détermination.

**28 bis.11.2** Si la ligne principale de détermination est la ligne 33, 34, 36, 38, 40, 41, 43, 44, 46, 48, 50, 51 ou 53, il est nécessaire que les propriétés additionnelles correspondent à celles spécifiées dans l'une des lignes 1 à 14. La ligne 1 à 14 qui s'applique est appelée la ligne de détermination de la date.

**28 bis.11.3** Si la ligne principale de détermination est la ligne 33, 35, 36, 39, 40, 42, 43, 45, 46, 49, 50, 52 ou 53, il est nécessaire que les propriétés additionnelles correspondent à celles spécifiées dans l'une des lignes 15 à 32. La ligne 15 à 32 qui s'applique est appelée la ligne de détermination de l'heure.

**28 bis.11.4** Dans les types **DATE-TYPE**, **TIME-TYPE** et **MIXED-ENCODING**, l'alternative **row-n** est sélectionnée si la ligne de détermination de la date, la ligne de détermination de l'heure ou la ligne principale de détermination (respectivement) est la ligne *n*.

**28 bis.11.5** Le codage de la valeur abstraite est le codage du type **MIXED-ENCODING**:

```
MIXED-ENCODING ::= CHOICE {
    row-1    CENTURY-ENCODING,
    row-2    ANY-CENTURY-ENCODING,
    row-3    YEAR-ENCODING,
    row-4    ANY-YEAR-ENCODING,
    row-5    YEAR-MONTH-ENCODING,
    row-6    ANY-YEAR-MONTH-ENCODING,
    row-7    DATE-ENCODING,
    row-8    ANY-DATE-ENCODING,
```

```

row-9  YEAR-DAY-ENCODING,
row-10 ANY-YEAR-DAY-ENCODING,
row-11 YEAR-WEEK-ENCODING,
row-12 ANY-YEAR-WEEK-ENCODING,
row-13 YEAR-WEEK-DAY-ENCODING,
row-14 ANY-YEAR-WEEK-DAY-ENCODING,
row-15 HOURS-ENCODING,
row-16 HOURS-UTC-ENCODING,
row-17 HOURS-AND-DIFF-ENCODING,
row-18 MINUTES-ENCODING,
row-19 MINUTES-UTC-ENCODING,
row-20 MINUTES-AND-DIFF-ENCODING,
row-21 TIME-OF-DAY-ENCODING,
row-22 TIME-OF-DAY-UTC-ENCODING,
row-23 TIME-OF-DAY-AND-DIFF-ENCODING,
row-24 HOURS-AND-FRACTION-ENCODING,
row-25 HOURS-UTC-AND-FRACTION-ENCODING,
row-26 HOURS-AND-DIFF-AND-FRACTION-ENCODING,
row-27 MINUTES-AND-FRACTION-ENCODING,
row-28 MINUTES-UTC-AND-FRACTION-ENCODING,
row-29 MINUTES-AND-DIFF-AND-FRACTION-ENCODING,
row-30 TIME-OF-DAY-AND-FRACTION-ENCODING,
row-31 TIME-OF-DAY-UTC-AND-FRACTION-ENCODING,
row-32 TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING,
row-33 DATE-TIME-ENCODING {DATE-TYPE, TIME-TYPE},
row-34 START-END-DATE-INTERVAL-ENCODING {DATE-TYPE},
row-35 START-END-TIME-INTERVAL-ENCODING {TIME-TYPE},
row-36 START-END-DATE-TIME-INTERVAL-ENCODING {DATE-TYPE, TIME-TYPE},
row-37 DURATION-INTERVAL-ENCODING,
row-38 START-DATE-DURATION-INTERVAL-ENCODING {DATE-TYPE},
row-39 START-TIME-DURATION-INTERVAL-ENCODING {TIME-TYPE},
row-40 START-DATE-TIME-DURATION-INTERVAL-ENCODING {DATE-TYPE, TIME-TYPE},
row-41 DURATION-END-DATE-INTERVAL-ENCODING {DATE-TYPE},
row-42 DURATION-END-TIME-INTERVAL-ENCODING {TIME-TYPE},
row-43 DURATION-END-DATE-TIME-INTERVAL-ENCODING {DATE-TYPE, TIME-TYPE},
row-44 REC-START-END-DATE-INTERVAL-ENCODING {DATE-TYPE},
row-45 REC-START-END-TIME-INTERVAL-ENCODING {TIME-TYPE},
row-46 REC-START-END-DATE-TIME-INTERVAL-ENCODING {DATE-TYPE, TIME-TYPE},
row-47 REC-DURATION-INTERVAL-ENCODING,
row-48 REC-START-DATE-DURATION-INTERVAL-ENCODING {DATE-TYPE},
row-49 REC-START-TIME-DURATION-INTERVAL-ENCODING {TIME-TYPE},
row-50 REC-START-DATE-TIME-DURATION-INTERVAL-ENCODING
      {DATE-TYPE, TIME-TYPE},
row-51 REC-DURATION-END-DATE-INTERVAL-ENCODING {DATE-TYPE},
row-52 REC-DURATION-END-TIME-INTERVAL-ENCODING {TIME-TYPE},
row-53 REC-DURATION-END-DATE-TIME-INTERVAL-ENCODING
      {DATE-TYPE, TIME-TYPE} }

```

où le codage du type de chaque alternative est spécifié dans le paragraphe identifié dans le Tableau 2, colonne 3 de la ligne principale de détermination.

**28 bis.11.6** Le type `DATE-TYPE` est défini comme:

```

DATE-TYPE ::= CHOICE {
row-1  CENTURY-ENCODING,
row-2  ANY-CENTURY-ENCODING,
row-3  YEAR-ENCODING,
row-4  ANY-YEAR-ENCODING,
row-5  YEAR-MONTH-ENCODING,
row-6  ANY-YEAR-MONTH-ENCODING,
row-7  DATE-ENCODING,
row-8  ANY-DATE-ENCODING,
row-9  YEAR-DAY-ENCODING,
row-10 ANY-YEAR-DAY-ENCODING,
row-11 YEAR-WEEK-ENCODING,
row-12 ANY-YEAR-WEEK-ENCODING,
row-13 YEAR-WEEK-DAY-ENCODING,
row-14 ANY-YEAR-WEEK-DAY-ENCODING }

```

où le codage du type de chaque alternative est spécifié dans le paragraphe identifié dans le Tableau 2, colonne 3 de la ligne de détermination de la date.

**28 bis.11.7** Le type **TIME-TYPE** est défini comme:

```

TIME-TYPE::= SEQUENCE {
  number-of-digits INTEGER (1..MAX) OPTIONAL,
  time-type CHOICE {
    row-15 HOURS-ENCODING,
    row-16 HOURS-UTC-ENCODING,
    row-17 HOURS-AND-DIFF-ENCODING,
    row-18 MINUTES-ENCODING,
    row-19 MINUTES-UTC-ENCODING,
    row-20 MINUTES-AND-DIFF-ENCODING,
    row-21 TIME-OF-DAY-ENCODING,
    row-22 TIME-OF-DAY-UTC-ENCODING,
    row-23 TIME-OF-DAY-AND-DIFF-ENCODING,
    row-24 HOURS-AND-FRACTION-ENCODING,
    row-25 HOURS-UTC-AND-FRACTION-ENCODING,
    row-26 HOURS-AND-DIFF-AND-FRACTION-ENCODING,
    row-27 MINUTES-AND-FRACTION-ENCODING,
    row-28 MINUTES-UTC-AND-FRACTION-ENCODING,
    row-29 MINUTES-AND-DIFF-AND-FRACTION-ENCODING,
    row-30 TIME-OF-DAY-AND-FRACTION-ENCODING,
    row-31 TIME-OF-DAY-UTC-AND-FRACTION-ENCODING,
    row-32 TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING } }

```

où le codage du type de chaque alternative est spécifié dans le paragraphe identifié dans le Tableau 2, colonne 3 de la ligne de détermination de l'heure.

**28 bis.11.8** Le composant **number-of-digits** est présent dans le type **TIME-TYPE** si et seulement si le composant **time-type** est l'une des alternatives **row-24** à **row-32**. Il code le nombre de chiffres dans la partie décimale de la valeur abstraite.





## SÉRIES DES RECOMMANDATIONS UIT-T

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