



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

X.694

(01/2004)

СЕРИЯ X: СЕТИ ПЕРЕДАЧИ ДАННЫХ
И ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

Сетевые и системные аспекты ВОС –
Система абстрактных синтаксических
нотаций Один (ASN.1)

**Информационные технологии –
правила кодирования ASN.1: отображение
определений схемы языка XML консорциума
W3C в системе обозначений ASN.1**

Рекомендация МСЭ-Т X.694

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ X

СЕТИ ПЕРЕДАЧИ ДАННЫХ И ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

СЕТИ ПЕРЕДАЧИ ДАННЫХ ОБЩЕГО ПОЛЬЗОВАНИЯ	
Службы и услуги	X.1–X.19
Интерфейсы	X.20–X.49
Передача, сигнализация и коммутация	X.50–X.89
Сетевые аспекты	X.90–X.149
Техническая эксплуатация	X.150–X.179
Административные предписания	X.180–X.199
ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ	
Модель и нотации	X.200–X.209
Определение услуг	X.210–X.219
Спецификации протоколов с соединениями	X.220–X.229
Спецификации протоколов без соединений	X.230–X.239
Формы заявок о соответствии реализации протокола (ЗСРП)	X.240–X.259
Идентификация протоколов	X.260–X.269
Протоколы безопасности	X.270–X.279
Объекты, административно управляемые уровнем	X.280–X.289
Аттестационное тестирование	X.290–X.299
ВЗАИМОДЕЙСТВИЕ МЕЖДУ СЕТЯМИ	
Общие положения	X.300–X.349
Спутниковые системы передачи данных	X.350–X.369
IP-сети	X.370–X.399
СИСТЕМЫ ОБРАБОТКИ СООБЩЕНИЯ	X.400–X.499
СПРАВОЧНАЯ СИСТЕМА	X.500–X.599
СЕТЕВЫЕ И СИСТЕМНЫЕ АСПЕКТЫ ВОС	
Сетевые аспекты	X.600–X.629
Эффективность	X.630–X.639
Качество обслуживания	X.640–X.649
Обозначение, адресация и регистрация	X.650–X.679
Система абстрактных синтаксических нотаций Один (ASN.1)	X.680–X.699
АДМИНИСТРАТИВНОЕ УПРАВЛЕНИЕ ВОС	
Порядок правления системой и архитектура	X.700–X.709
Управление услугой передачи и протокол	X.710–X.719
Структура управленческой информации	X.720–X.729
Функции управления и функции ODMA	X.730–X.799
БЕЗОПАСНОСТЬ	X.800–X.849
ПРИЛОЖЕНИЯ ВОС	
Исполнение, совмещение и восстановление	X.850–X.859
Обработка транзакций	X.860–X.879
Удаленные операции	X.880–X.899
ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА	X.900–X.999

Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.

**Информационные технологии – Правила кодирования ASN.1:
Отображение определений схемы языка XML консорциума W3C
в системе обозначений ASN.1**

Резюме

В данной Рекомендации | Международном стандарте определяются правила отображения схемы XSD (схемы, отвечающей спецификации схемы языка XML консорциума W3C) в схеме ASN.1, чтобы использовать правила кодирования системы обозначений ASN.1, такие как базовые правила кодирования (BER), отличительные правила кодирования (DER), правила пакетного кодирования (PER) или правила кодирования языка XML (XER) для передачи информации, определяемой схемой XSD.

Использование данной Рекомендации | Международного стандарта с расширенными правилами кодирования языка XML в системе обозначений ASN.1 (EXTENDED-XER) обеспечивает то же представление значений в языке XML, что и представление, определяемое исходной схемой XSD.

Источник

Рекомендация МСЭ-Т X.694 утверждена 13 января 2004 года 17-й Исследовательской комиссией МСЭ-Т (2001–2004 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т A.8. Идентичный текст также опубликован как стандарт ИСО/МЭК 8825-5.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

Всемирная ассамблея по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяет темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соответствие положениям данной Рекомендации является добровольным делом. Однако в Рекомендации могут содержаться определенные обязательные положения (для обеспечения, например, возможности взаимодействия или применимости), и тогда соответствие данной Рекомендации достигается в том случае, если выполняются все эти обязательные положения. Для выражения требований используются слова "shall" ("следует", "обязан") или некоторые другие обязывающие термины, такие как "must" ("должен"), а также их отрицательные эквиваленты. Использование таких слов не предполагает, что соответствие данной Рекомендации требуется от каждой стороны.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на то, что практическое применение или реализация этой Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, обоснованности или применимости заявленных прав интеллектуальной собственности, независимо от того, отстаиваются ли они членами МСЭ или другими сторонами вне процесса подготовки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ не получил извещения об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для реализации этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ.

© ITU 2005

Все права сохранены. Никакая часть данной публикации не может быть воспроизведена с помощью каких-либо средств без предварительного письменного разрешения МСЭ.

СОДЕРЖАНИЕ

	<i>Стр.</i>
1 Область применения	1
2 Нормативные ссылки.....	1
2.1 Идентичные Рекомендации Международные стандарты	1
2.2 Дополнительные ссылки.....	2
3 Определения.....	2
3.1 Импортируемые определения	2
3.2 Дополнительные определения	3
4 Сокращения	3
5 Система обозначений	3
6 Назначение и степень стандартизации	3
7 Отображение схем XSD	4
8 Игнорируемые схемные компоненты и свойства	5
9 Модуль ASN.1 и пространства имен.....	6
10 Преобразование имен	6
10.1 Общие положения	6
10.2 Генерирование определений типов ASN.1, являющихся ссылками на присвоения типов ASN.1	7
10.3 Генерирование идентификаторов и имен ссылок на типы	7
10.4 Порядок отображения	9
11 Отображение использования встроенных типов данных XSD.....	10
12 Отображение аспектов.....	11
12.1 Аспекты length, minLength и maxLength	11
12.2 Аспект pattern.....	11
12.3 Аспект whiteSpace.....	11
12.4 Аспект enumeration	12
12.5 Прочие аспекты	14
13 Отображение компонент simple type definition (определение простого типа)	14
14 Отображение компонент element declaration (описание элемента).....	16
15 Отображение компонент attribute declaration (описание атрибута).....	17
16 Отображение значений компонент simple type definition (определение простого типа).....	17
17 Отображение компонент model group definition (определение группы моделей).....	17
18 Отображение компонент model group (группа моделей)	17
19 Отображение компонент particle (частица).....	17
20 Отображение компонент complex type definition (определение сложного типа)	19
21 Отображение компонент wildcard (метасимвол).....	20
22 Отображение компонент attribute use (использование атрибута)	20
23 Отображение использований компонент simple type definition (определение простого типа) и complex type definition (определение сложного типа)	21
24 Отображение особых использований компонент simple type definition и complex type definition (заменяемых)	22
25 Отображение особых использований компонент simple и complex type definitions (substitutable, nillable).....	23
26 Отображение особых использований компонент simple type definitions (nillable)	24
27 Отображение особых использований компонент complex type definitions (nillable)	24
28 Отображение особых использований компонент element declarations (головной элемент группы замены элементов)	25
29 Генерирование особых присвоений типов ASN.1 для компонент element declarations	26
30 Генерирование особых присвоений типов ASN.1 для компонент type definitions	27
31 Генерирование особых присвоений типов ASN.1 для групп замены элементов	27
Приложение А – Определение типов ASN.1, соответствующих встроенным типам данных XSD	28
Приложение В – Присвоение значений идентификаторов объектов.....	33

	<i>Стр.</i>
Приложение С – Примеры отображений.....	34
С.1 Схема, использующая компоненты simple type definitions (определение простого типа)....	34
С.2 Соответствующие определения ASN.1	35
С.3 Дальнейшие примеры	36
С.3.1 Схемные документы с информационными единицами элементов import и include..	36
С.3.2 Отображение компонент simple type definition (определение простого типа)	37
С.3.3 Отображение компонент facet (аспектов)	38
С.3.4 Отображение компонент element declarations	41
С.3.5 Отображение компонент attribute use (использование атрибута) и attribute declaration (описание атрибута).....	46
С.3.6 Отображение компонент model group definition (определение группы моделей).....	47
С.3.7 Отображение компонент particle (частица).....	48
С.3.8 Отображение компонент complex type definitions.....	49
С.3.9 Отображение атрибутов wildcard (метасимвол)	54
Приложение D – Использование отображения для обеспечения двоичных кодирований для схемы на языке XML консорциума W3C	57
D.1 Кодирование схем XSD	57
D.2 Передача без использования схемы XSD для схем	57
D.3 Передача с использованием схемы XSD для схем	57

Введение

В данной Рекомендации | Международном стандарте дано описание отображения определения схемы языка XML консорциума W3C (схемы XSD) в схеме ASN.1. Это отображение может быть применено к любой схеме XSD. Оно точно определяет генерирование одного или нескольких модулей системы обозначений ASN.1, содержащих определения типов, вместе с командами кодирования XER системы обозначений ASN.1. Совместно они именуются схемой системы обозначений ASN.1 для документов на языке XML.

Эта схема ASN.1 при использовании с расширенными правилами кодирования языка XML в системе обозначений ASN.1 (EXTENDED-XER) может быть использована для генерирования и проверки правильности того же множества документов версии 1.0 языка XML консорциума W3C, что и исходная схема XSD. Результирующие типы и виды кодирования системы обозначений ASN.1 поддерживают то же семантическое содержание, что и схема XSD. Таким образом, средства системы обозначений ASN.1 могут быть использованы взаимозаменяемым образом со средствами XSD для генерирования и обработки определяемых документов на языке XML.

В сочетании с данным стандартизированным отображением могут быть использованы другие стандартизированные правила кодирования в системе обозначений ASN.1, такие как отличительные правила кодирования (DER) или правила пакетного кодирования (PER).

Сочетание данной Рекомендации | Международного стандарта с правилами кодирования в системе обозначений ASN.1 обеспечивает полностью стандартизированное и независимое от поставщика компактное двоичное кодирование канонического вида для данных, определяемых при использовании схемы XSD.

Схема системы обозначений ASN.1 обеспечивает четкое разделение между спецификацией информационного содержимого сообщений (их абстрактным синтаксисом) и точной формой документа на языке XML (например, при использовании атрибутов вместо элементов). Результатом этого является как более ясная, так и менее подробная в целом схема, чем исходная схема XSD.

Приложение А является неотъемлемой частью настоящей Рекомендации | Международного стандарта и является модулем в системе обозначений ASN.1, содержащим множество присвоений типов ASN.1, соответствующих каждому из встроенных типов данных XSD. Отображения схем XSD в схемах ASN.1 либо импортируют имена ссылок на тип этих присвоений типов, либо включают встраивание определений типов.

Приложение В не является неотъемлемой частью настоящей Рекомендации | Международного стандарта, и в нем кратко определяются значения идентификаторов объектов, присвоенные в данной Рекомендации | Международном стандарте.

Приложение С не является неотъемлемой частью настоящей Рекомендации | Международного стандарта и содержит примеры отображения схем XSD в схемах ASN.1.

Приложение D не является неотъемлемой частью настоящей Рекомендации | Международного стандарта и содержит описания по использованию отображения, определяемого в настоящей Рекомендации | Международном стандарте, в сочетании со стандартизированными правилами кодирования в системе обозначений ASN.1 для обеспечения компактного и канонического кодирования данных, определяемых при использовании схемы XSD.

МЕЖДУНАРОДНЫЙ СТАНДАРТ РЕКОМЕНДАЦИЯ МСЭ-Т

Информационные технологии – Правила кодирования ASN.1: Отображение определений схемы языка XML консорциума W3C в системе обозначений ASN.1

1 Область применения

В данной Рекомендации | Международном стандарте дано описание отображения любой схемы XSD в схеме системы обозначений ASN.1. Схема ASN.1 поддерживает ту же семантику и проверяет правильность того же множества документов на языке XML.

В данной Рекомендации | Международном стандарте дается описание конечных команд кодирования XER, которые должны применяться как часть определяемого отображения в типах системы обозначений ASN.1, но не дается описания того, какая синтаксическая форма должна использоваться для описания этих конечных команд кодирования XER или порядка и способа их присвоения.

ПРИМЕЧАНИЕ. – Разработчики средств генерирования этих отображений могут выбирать любую синтаксическую форму или порядок присвоения, который приводит к применению определяемых конечных команд кодирования XER. В примерах, приведенных в данной Рекомендации | Международном стандарте, как правило, используется префиксная форма типа, но использование управляющей секции кодирования XER может оказаться предпочтительным для отображения полной схемы XSD в отношении стиля.

Имеются различные способы синтаксического присвоения команд кодирования XER для использования в кодировании с расширенными правилами (EXTENDED-XER) (например, использование команд префиксного кодирования типов ASN.1 или использование управляющей секции кодирования XER). Выбор этих синтаксических форм является вопросом стиля и выходит за рамки настоящей Рекомендации | Международного стандарта.

2 Нормативные ссылки

Указанные ниже Рекомендации | Международные стандарты и спецификации консорциума W3C содержат положения, которые путем ссылки на них в данном тексте составляют положения настоящей Рекомендации | Международного стандарта. На момент публикации указанные издания были действующими. Все Рекомендации, Международные стандарты и спецификации консорциума W3C могут подвергаться пересмотру; поэтому сторонам соглашений, основанных на настоящей Рекомендации | Международном стандарте, предлагается изучить возможность применения последнего издания перечисленных ниже Рекомендаций, Международных стандартов и спецификаций консорциума W3C. Члены МЭК и ИСО ведут регистры действующих в настоящее время Международных стандартов. Бюро стандартизации электросвязи МСЭ ведет список действующих в настоящее время Рекомендаций МСЭ-Т. Консорциум W3C ведет список действующих в настоящее время спецификаций консорциума W3C. Ссылка на документ в данной Рекомендации | Международном стандарте не придает ему как отдельному документу статус Рекомендации или Международного стандарта.

2.1 Идентичные Рекомендации | Международные стандарты

ПРИМЕЧАНИЕ. – Ниже перечислен полный набор Рекомендаций | Международных стандартов по системе обозначений ASN.1, поскольку они все могут быть применены при отдельных использованиях данной Рекомендации | Международного стандарта. Если в тексте настоящей Рекомендации | Международного стандарта отсутствуют прямые ссылки на эти источники, то к ссылке добавляется символ †.

- ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.
- ITU-T Recommendation X.681 (2002) | ISO/IEC 8824-2:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.
- ITU-T Recommendation X.682 (2002) | ISO/IEC 8824-3:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.
- ITU-T Recommendation X.683 (2002) | ISO/IEC 8824-4:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications*.
- ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER)*.
- ITU-T Recommendation X.691 (2002) | ISO/IEC 8825-2:2002, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

- ITU-T Recommendation X.692 (2002) | ISO/IEC 8825-3:2002, *Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN)*.
- ITU-T Recommendation X.693 (2001) | ISO/IEC 8825-4:2002, *Information technology – ASN.1 encoding rules: XML Encoding Rules (XER)*.
- ITU-T Recommendation X.693 (2001)/Amd.1 (2003) | ISO/IEC 8825-4:2002/Amd.1:2004, *Information technology – ASN.1 encoding rules: XML Encoding Rules (XER) – Amendment 1: XER Encoding Instructions and EXTENDED-XER*.

2.2 Дополнительные ссылки

- ISO 8601:2000, *Data elements and interchange formats – Information interchange – Representation of dates and times*.
 - ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.
 - W3C XML 1.0:2000, *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation, Copyright © [6 October 2000] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2000/REC-xml-20001006>.
 - W3C XML Namespaces:1999, *Namespaces in XML*, W3C Recommendation, Copyright © [14 January 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/1999/REC-xml-names-19990114>.
 - W3C XML Information Set:2001, *XML Information Set*, W3C Recommendation, Copyright © [24 October 2001] World Wide Web Consortium (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2001/REC-xml-info-set-20011024>.
 - W3C XML Schema:2001, *XML Schema Part 1: Structures*, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>.
 - W3C XML Schema:2001, *XML Schema Part 2: Datatypes*, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502>.
- ПРИМЕЧАНИЕ. – Когда в данной Рекомендации | Международном стандарте используется сноска "Схема языка XML консорциума W3C", это относится к части 1 схемы языка XML консорциума W3C и части 2 схемы языка XML консорциума W3C.
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.
 - IETF RFC 1766 (1995), *Tags for the Identification of Languages*.

3 Определения

3.1 Импортируемые определения

3.1.1 В настоящей Рекомендации | Международном стандарте используются термины, определение которых дано в Рекомендации МСЭ-Т X.680 | ИСО/МЭК 8824-1 и Рекомендации МСЭ-Т X.693 | ИСО/МЭК 8825-4.

ПРИМЕЧАНИЕ. – В частности, термины "конечные команды кодирования XER", "префикс типа" и "управляющая секция кодирования XER" определены в упомянутых выше Рекомендациях | Международных стандартах.

3.1.2 В настоящей Рекомендации | Международном стандарте также используются термины, которые определены в схеме языка XML консорциума W3C и в информационном множестве языка XML консорциума W3C.

ПРИМЕЧАНИЕ 1. – Считается, что эти термины не противоречат терминам, на которые делаются ссылки в подразделе 3.1.1. Если такое противоречие имеет место, то применяется определение термина из подраздела 3.1.1.

ПРИМЕЧАНИЕ 2. – В частности, термины "компонента схемы" и "свойство (компоненты схемы)" определены в схеме языка XML консорциума W3C, а термин "единица информации элемента" определен в информационном множестве языка XML консорциума W3C.

3.2 Дополнительные определения

Для целей настоящей Рекомендации | Международного стандарта применяются следующие дополнительные определения:

3.2.1 Пространство имен XSD: Пространство имен с идентификатором URI сайта "http://www.w3.org/2001/XMLSchema".

3.2.2 Пространство имен XSI: Пространство имен с идентификатором URI сайта "http://www.w3.org/2001/XMLSchema-instance".

3.2.3 Пространство имен XML: Пространство имен с идентификатором URI сайта "http://www.w3.org/XML/1998/namespace".

4 Сокращения

Для целей настоящей Рекомендации | Международного стандарта применяются следующие сокращения:

ASN.1	Система абстрактных синтаксических нотаций Один
BER	Базовые правила кодирования (ASN.1)
DER	Отличительные правила кодирования (ASN.1)
PER	Правила пакетного кодирования (ASN.1)
URI	Унифицированный идентификатор ресурсов (IETF)
XER	Правила кодирования языка XML (ASN.1)
XML	Расширяемый маркировочный язык (W3C)
XSD	Схема языка XML (W3C)

5 Система обозначений

5.1 В настоящей Рекомендации | Международном стандарте делаются ссылки на систему обозначений, определяемую в Рекомендации МСЭ-Т X.680 | ИСО/МЭК 8824-1, Рекомендации МСЭ-Т X.682 | ИСО/МЭК 8824-3, в версии 1.0 языка XML консорциума W3C и в схеме языка XML консорциума W3C.

5.2 Когда в тексте настоящей Рекомендации | Международного стандарта необходимо описать либо формально, либо на примерах присвоение команд кодирования согласно правилам XER, обычно используется система обозначений с префиксами типов (см. подразделы 6.3 и 6.4). В Приложении А используется управляющая секция кодирования XER.

5.3 В настоящей Рекомендации | Международном стандарте для системы обозначений ASN.1 используется жирный шрифт *Courier*, а для системы обозначений XSD и терминов и понятий XSD используется жирный шрифт *Arial*.

5.4 В схемах XSD в примерах, приведенных в данной Рекомендации | Международном стандарте, используется префикс **xsd**: для определения пространства имен XSD.

6 Назначение и степень стандартизации

6.1 Отображение в ASN.1, описанное в данной Рекомендации | Международном стандарте, гарантирует, что:

- все результирующие модули ASN.1, генерируемые средствами, отвечающими данной Рекомендации | Международному стандарту (из той же схемы XSD), определяют те же (структурированные) абстрактные значения;
- все правила кодирования типа BASIC-XER, CXER, EXTENDED-XER и двоичные кодирования этой результирующей спецификации ASN.1 будут порождать те же виды кодирования (в зависимости от вариантов кодера); и
- все документы на языке XML, которые отвечают схеме XSD исходного текста, являются действительными кодированиями по правилам EXTENDED-XER абстрактных значений этой спецификации ASN.1.

6.2 Имеется много аспектов определения ASN.1 (таких как использование пробела, кодирование управляющих секций или префиксов типов), которые не влияют ни на абстрактные определяемые значения, ни на кодирования по правилам XER или двоичные кодирования этих значений. Такие аспекты определения ASN.1, вообще говоря, в данной Рекомендации | Международном стандарте не стандартизированы.

6.3 В ASN.1 имеется много разных способов присвоения команды кодирования XER типу, включая:

- использование префикса типа для каждой присвоенной команды кодирования; или
- использование управляющей секции кодирования с отдельной командой кодирования для каждого требуемого присвоения; или
- использование управляющей секции кодирования с одной командой кодирования, выполняющей глобальное присвоение, возможно, дополненной использованием командой отрицания кодирования для конкретных типов.

6.4 В данной Рекомендации | Международном стандарте определяется, когда будет присутствовать конечная команда кодирования XER, и в большинстве примеров используется синтаксис подраздела 6.3 а).

Однако использование разных вариантов в подразделе 6.3 не стандартизировано, и в соответствующих реализациях отображения может выбираться любая синтаксическая форма (или сочетание синтаксических форм) для присвоения конечных команд кодирования XER.

ПРИМЕЧАНИЕ. – Осуществление выбора среди этих вариантов не влияет на конечное двоичное кодирование или кодирование на языке XML.

6.5 Формальная спецификация требуемого отображения не предоставляется.

6.6 Данная Рекомендация | Международный стандарт относится только к отображению схем XSD, которые соответствуют схеме языка XML консорциума W3C.

ПРИМЕЧАНИЕ. – Такое соответствие может быть достигнуто либо предоставлением одного или нескольких документов схемы XSD консорциума W3C, либо другими средствами, как описано в схеме языка XML консорциума W3C.

7 Отображение схем XSD

7.1 Отображение базируется на схеме XSD исходного текста, которая является множеством схемных компонент (см. часть 1, 2.2 Схемы языка XML консорциума W3C). Для этого отображения не требуется или не предполагается отдельное представление схемных компонент или множеств схемных компонент, хотя ожидается, что схема XSD исходного текста будет обычно предоставляться как один или несколько схемных документов на языке XML (см. часть 1, подраздел 3.15.2 Схемы языка XML консорциума W3C).

ПРИМЕЧАНИЕ 1. – Поскольку отображение определено в терминах схемных компонент (а не в терминах их представления на языке XML), то на него не влияют детали представления на языке XML, такие как использование множества схемных документов, связанных информационными единицами элементов **xsd:include** и **xsd:redefine**, помещение информационных единиц элементов в один или другой схемный документ, порядок информационных единиц элемента **xsd:attribute** в информационной единице элемента **xsd:complexType** и так далее.

ПРИМЕЧАНИЕ 2. – Два множества схемных документов, которые отличаются по многим аспектам, но представляют то же множество схемных компонент, генерируют то же множество присвоений типов ASN.1 с теми же конечными командами кодирования, присвоенными им и их компонентам, на любую глубину.

7.2 Схема XSD исходного текста должна отвечать всем ограничениям, налагаемым спецификацией XSD. Если схема XSD исходного текста (частично или вся) представлена как множество схемных документов на языке XML, то каждый схемный документ должен быть действительным согласно схеме XSD для схем (см. часть 1, Добавление А Схемы языка XML консорциума W3C).

7.3 По крайней мере один модуль ASN.1 (см. подраздел 7.4) должен генерироваться для каждой отличной компоненты **target namespace** (либо имя пространства имен, либо ключевое слово **absent** (отсутствует)), которая является компонентой **target namespace** одной или нескольких схемных компонент в схеме XSD исходного текста. Каждый модуль ASN.1 должен содержать одно или несколько присвоений типов, соответствующих компонентам схемы верхнего уровня (см. подраздел 7.9), которые имеют одну и ту же компоненту **target namespace**. Каждый модуль ASN.1 может также содержать одно или несколько специальных присвоений типов ASN.1, чьи связанные с ними присвоения типов ASN.1 находятся в том же модуле ASN.1 (см. 7.6).

ПРИМЕЧАНИЕ. – Схемные компоненты, представленные во множестве схемных документов, становятся частью той же схемы XSD в результате использования информационных единиц элементов **xsd:include**, **xsd:redefine** и **xsd:import**.

7.4 Число модулей ASN.1, генерируемых для каждого компонента **target namespace** (включая ключевое слово **absent**), может быть больше одного, но каждый модуль ASN.1 не должен содержать присвоений типов, соответствующих схемным компонентам верхнего уровня с различными компонентами **target namespaces** (включая ключевое слово **absent**).

7.5 Когда множество модулей ASN.1 генерируется для данной компоненты **target namespace** (включая ключевое слово **absent**), все присвоения типов, содержащиеся в них, должны генерироваться, как если бы они добавлялись к одному модулю ASN.1 в целях генерирования различных имен ссылок на типы (см. подраздел 10.3). Имена ссылок на типы, генерируемые из компонент **name** схемных компонент верхнего уровня с данной компонентой **target namespace**, должны быть теми же именами ссылок на типы вне зависимости от числа модулей ASN.1, генерируемых для этой компоненты **target namespace**, и независимо от способа разделения присвоений типов среди различных модулей ASN.1.

ПРИМЕЧАНИЕ. – Это осуществляется для обеспечения гибкости не в ущерб взаимодействию.

7.6 Каждое специальное присвоение типа ASN.1 (см. разделы 29, 30 и 31) должно вводиться в тот же модуль ASN.1, что и связанное с ним присвоение типа ASN.1 (см., соответственно, подразделы 29.4, 31.4 и 30.4).

7.7 Все модули ASN.1, генерируемые отображением, должны содержать (в управляющей секции кодирования XER) команду кодирования **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** и команду кодирования **GLOBAL-DEFAULTS CONTROL-NAMESPACE**, описывающие пространство имен XSI.

7.8 Схема XSD исходного текста должна обрабатываться следующим образом:

- для каждой компоненты **element declaration** верхнего уровня присвоение типа ASN.1 должно генерироваться путем применения раздела 14 к компоненте **element declaration**;
- для каждой компоненты **attribute declaration** верхнего уровня присвоение типа ASN.1 должно генерироваться путем применения раздела 15 к компоненте **attribute declaration**;
- для каждой определяемой пользователем компоненты **simple type definition** на верхнем уровне присвоение типа ASN.1 должно генерироваться путем применения раздела 13 к компоненте **simple type definition**;
- для каждой компоненты **complex type definition** верхнего уровня должно генерироваться присвоение типа ASN.1 путем применения раздела 20 к компоненте **complex type definition**;

- е) для каждой компоненты **model group definition**, компонента которой **model group** имеет компоненту **compositor** компонент **sequence** или **choice**, присвоение типа ASN.1 должно генерироваться путем применения раздела 17 к компоненте **model group definition**.

ПРИМЕЧАНИЕ 1. – Остальные схемные компоненты схемы XSD исходного текста должны обрабатываться как результат отображения этих схемных компонент.

ПРИМЕЧАНИЕ 2. – Порядок, согласно которому должны отображаться схемные компоненты, описан в подразделе 10.4. Порядок элементов списка выше не имеет значения для отображения.

7.9 В столбце 1 таблицы 1 перечисляются схемные компоненты. В столбце 2 даются ссылки на разделы в схеме языка XML консорциума W3C, где определяется схемная компонента. В столбце 3 перечисляются разделы, где определяются отображения этих схемных компонент в ASN.1.

Таблица 1 – Отображение схемных компонент XSD

Схемные компоненты XSD	Ссылка на схему языка XML консорциума W3C	Отображение определяется в
attribute declaration	Часть 1, 3.2	разделе 15
element declaration	Часть 1, 3.3	разделе 14
complex type definition	Часть 1, 3.4	разделе 20
attribute use	Часть 1, 3.5	разделе 22
attribute group definition	Часть 1, 3.6	<i>не отображается как таковая</i>
model group definition	Часть 1, 3.7	разделе 17
model group	Часть 1, 3.8	разделе 18
particle	Часть 1, 3.9	разделе 19
wildcard	Часть 1, 3.10	разделе 21
identity-constraint definition	Часть 1, 3.11	<i>игнорируется отображением</i>
notation declaration	Часть 1, 3.12	<i>игнорируется отображением</i>
annotation	Часть 1, 3.13	<i>игнорируется отображением</i>
simple type definition	Часть 1, 3.14	разделах 11, 13
schema	Часть 1, 3.15	разделе 9
ordered	Часть 2, 4.2.2.1	<i>игнорируется отображением</i>
bounded	Часть 2, 4.2.3.1	<i>игнорируется отображением</i>
cardinality	Часть 2, 4.2.4.1	<i>игнорируется отображением</i>
numeric	Часть 2, 4.2.5.1	<i>игнорируется отображением</i>
length	Часть 2, 4.3.1.1	разделе 12
minLength	Часть 2, 4.3.2.1	разделе 12
maxLength	Часть 2, 4.3.3.1	разделе 12
pattern	Часть 2, 4.3.4.1	разделе 12
enumeration	Часть 2, 4.3.5.1	разделе 12
whiteSpace	Часть 2, 4.3.6.1	разделе 12
maxInclusive	Часть 2, 4.3.7.1	разделе 12
maxExclusive	Часть 2, 4.3.8.1	разделе 12
minExclusive	Часть 2, 4.3.9.1	разделе 12
minInclusive	Часть 2, 4.3.10.1	разделе 12
totalDigits	Часть 2, 4.3.11.1	разделе 12
fractionDigits	Часть 2, 4.3.12.1	разделе 12

8 Игнорируемые схемные компоненты и свойства

8.1 При отображении должны игнорироваться схемные компоненты и свойства, перечисленные в данном разделе.

8.2 Все компоненты **annotations** (см. часть 1, 3.13 Схемы языка XML консорциума W3C) должны игнорироваться.

ПРИМЕЧАНИЕ. – Все информационные единицы атрибутов в документе схемы с именами, квалифицированными пространствами имен, отличными от пространства имен XSD (см. часть 1, 3.13.1 Схемы языка XML консорциума W3C), являются свойством компонент **annotation** и игнорируются.

8.3 Все компоненты **identity-constraint definitions** (см. часть 1, 3.11 Схемы языка XML консорциума W3C) должны игнорироваться.

ПРИМЕЧАНИЕ. — Компонента **identity-constraint definition** предоставляет механизмы для определения ссылочных ограничений, которые могут быть необходимы в действительном экземпляре. В настоящее время в ASN.1 нет концепции таких ограничений, и такие ограничения не могут быть отображены в формальной спецификации ASN.1, но они могут быть включены как нормативные замечания, которые имеют обязательную силу при реализации приложения.

8.4 Все компоненты **notation declarations** (см. часть 1, 3.12 Схемы языка XML консорциума W3C) должны игнорироваться.

8.5 Все схемные компоненты, которые являются компонентами **fundamental facets** (**ordered**, **bounded**, **cardinality**, **numeric**) компонент **simple type definitions** (см. часть 2, 4.2 Схемы языка XML консорциума W3C), должны игнорироваться.

8.6 Свойства **identity-constraint definitions**, **substitution group exclusion** и **disallowed substitution** компонент **element declaration** должны игнорироваться.

8.7 Свойства **final**, **abstract** и **prohibited substitutions** компонент **complex type definition** должны игнорироваться.

8.8 Свойство **process contents** компонент **wildcard** должно игнорироваться.

ПРИМЕЧАНИЕ. — В ASN.1 не существует поддержки для какого-либо действия, кроме действия **skip**.

8.9 Свойства **fundamental facet** и **final** компонент **simple type definition** должны игнорироваться.

8.10 Все компоненты **value constraint**, которые присутствуют в каких-либо компонентах **element declaration** или **attribute declaration**, чья компонента **type definition** либо **xsd:QName**, либо компонента **simple type definition**, полученная из **xsd:QName** или **xsd:NOTATION**, должны игнорироваться.

8.11 Все компоненты **attribute group definition** должны игнорироваться.

ПРИМЕЧАНИЕ. — Компоненты **attribute use** в компоненте **attribute group definition** становятся частью компонент **attribute use** в компонентах **complex type definition**, чье представление на языке XML содержит ссылку на компоненту **attribute group definition**.

9 Модуль ASN.1 и пространства имен

ПРИМЕЧАНИЕ. — Полное описание связи между понятием пространства имен в пространствах имен языка XML консорциума W3C и именованием приведено в ASN.1 в Рекомендации МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 16. Именам ссылок на типы и идентификаторам, определяемым в модуле ASN.1, присваивается пространство имен посредством команды кодирования **NAMESPACE**, в остальных случаях они лишены пространства имен. Это отображение по мере необходимости генерирует команды кодирования **NAMESPACE**.

9.1 Отображение генерирует один или несколько модулей ASN.1, соответствующих всем схемным компонентам в схеме, которая имеет ту же компоненту **target namespace**.

9.2 Элемент "ModuleIdentifier" (Идентификатор модуля) ASN.1 (см. Рекомендацию МСЭ-Т X.680 | ИСО/МЭК 8824-1, раздел 12), генерируемый отображением, не стандартизирован. Там, где используются операторы **IMPORTS**, имена модулей и идентификаторы модулей ASN.1 в операторах **IMPORTS** должны быть именами и идентификаторами модулей, генерируемыми для модулей ASN.1, генерируемых отображением.

ПРИМЕЧАНИЕ. — Выбор элемента "ModuleIdentifier" не влияет на кодирование в каком-либо из стандартных правил кодирования.

9.3 Модули ASN.1 должны иметь элемент "TagDefault" ("Тег по умолчанию") из атрибута **AUTOMATIC TAGS**.

9.4 В каждом модуле ASN.1 должен иметься оператор ASN.1 **IMPORTS**, импортирующий имена ссылок на типы ASN.1 в модуле, именуемом **XSD**, описанном в Приложении А, на которые ссылаются в модуле ASN.1.

9.5 Оператор **IMPORTS** должен также импортировать имена ссылок на типы ASN.1 присвоений типов, которые были помещены (в результате отображения) в другие модули ASN.1, но на которые делаются ссылки в данном модуле ASN.1.

9.6 Оператор **EXPORTS** должен отсутствовать.

ПРИМЕЧАНИЕ. — Это означает, что все имена ссылок на типы ASN.1 в модуле ASN.1 могут быть импортированы в другие модули.

10 Преобразование имен

10.1 Общие положения

10.1.1 В данной Рекомендации | Международном стандарте описывается генерирование:

- имен ссылок на типы ASN.1, соответствующих компонентам **name** компонент **model group definition**, компонент **element declaration** на верхнем уровне, компонент **attribute declaration** на верхнем уровне, компонент **complex type definition** на верхнем уровне и определяемых пользователем компонент **simple type definition** на верхнем уровне;
- идентификаторов ASN.1, соответствующих компонентам **name** компонент **element declaration** на верхнем уровне, компонент **attribute declaration** на верхнем уровне, локальных компонент **element declaration** и локальных компонент **attribute declaration**;
- идентификаторов ASN.1 для отображения определенных компонент **simple type definition** с аспектом **enumeration** (см. подразделы 12.4.1 и 12.4.2);

- d) имен ссылок на типы ASN.1 присвоений специального типа (см. разделы 29, 30 и 31); и
- e) идентификаторов ASN.1 определенных компонент последовательности, введенных отображением (см. раздел 20).

10.1.2 Все эти имена ASN.1 генерируются путем применения положений подраздела 10.3 либо к компоненте **name** соответствующей схемной компоненты, к члену компоненты **value** аспекта **enumeration** или к определенной строке символов, описанных в соответствующих разделах данной Рекомендации | Международного стандарта.

10.2 Генерирование определений типов ASN.1, являющихся ссылками на присвоения типов ASN.1

10.2.1 Данный подраздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, являющегося ссылкой ("DefinedType") ("Определяемый тип") на присвоение типа ASN.1.

10.2.2 Если определение типа ASN.1 (например, R), которое является "Определяемым типом", подлежит вводу в модуль ASN.1 (например, M), иной, чем модуль ASN.1, где вводится присвоение типа ASN.1 (например, TA), на которое делается ссылка, и имя ссылки на тип для TA идентично либо имени ссылки на тип другого присвоения типа ASN.1, вводимого в модуль M, либо другому имени ссылки на тип, импортируемому в модуль M, тогда R должен быть членом "ExternalTypeReference" (Внешняя ссылка на тип) (образуемая по мере необходимости для модуля M) для TA; в противном случае она должна быть элементом "typereference" (ссылка на тип) для TA.

10.3 Генерирование идентификаторов и имен ссылок на типы

10.3.1 Этот подраздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования имени ссылки на тип или идентификатора ASN.1.

10.3.2 Компоненты **name** компонент **attribute declaration**, **element declaration**, **model group definition**, определяемых пользователем компонент **simple type definition** на верхнем уровне и компонент **complex type definition** на верхнем уровне, могут быть идентичны резервируемым словам ASN.1 или могут содержать символы, недопустимые в идентификаторах ASN.1 или именах ссылок на типы ASN.1. Кроме того, имеются случаи, когда требуется, чтобы имена ASN.1 были различными, где допускается, чтобы компоненты **name** соответствующих схемных компонент XSD (от которых отображаются имена ASN.1) были идентичны.

10.3.3 Должны применяться следующие преобразования по порядку к каждой строке символов, отображаемой в имени ASN.1, где каждое преобразование (исключая первое) применяется к результату предыдущего преобразования:

- Символы " " (ПРОБЕЛ), "." (ПОЛНЫЙ ОСТАНОВ) и "_" (НИЖНЯЯ ЛИНИЯ) должны все заменяться на "-" (ДЕФИС-МИНУС); и
- любой символ, за исключением от "A" до "Z" (от ЛАТИНСКОЙ ПРОПИСНОЙ БУКВЫ А до ЛАТИНСКОЙ ПРОПИСНОЙ БУКВЫ Z), от "a" до "z" (от ЛАТИНСКОЙ СТРОЧНОЙ БУКВЫ А до ЛАТИНСКОЙ СТРОЧНОЙ БУКВЫ Z), от "0" до "9" (от ЦИФРЫ НОЛЬ до ЦИФРЫ ДЕВЯТЬ) и "-" (ДЕФИС-МИНУС), будет удален; и
- последовательность из двух и более символов "ДЕФИС-МИНУС" должна заменяться одним символом "ДЕФИС-МИНУС";
- символы "ДЕФИС-МИНУС", встречающиеся в начале или в конце имени, должны устраняться; и
- если строка символов, подлежащая использованию как имя ссылки на тип, начинается со строчной буквы, то первая буква должна быть прописной (переведена в верхний регистр); если строка символов начинается с цифры (от ЦИФРЫ НОЛЬ до ЦИФРЫ ДЕВЯТЬ), она должна иметь в качестве префикса символ "x" (ЛАТИНСКАЯ ПРОПИСНАЯ БУКВА X); и
- если строка символов, подлежащая использованию как идентификатор, начинается с прописной буквы, то первая буква должна быть сделана строчной (переведена в нижний регистр); если строка символов начинается с цифры (от ЦИФРЫ НОЛЬ до ЦИФРЫ ДЕВЯТЬ), она должна быть снабжена префиксом "x" (ЛАТИНСКАЯ СТРОЧНАЯ БУКВА X); и
- если строка символов, подлежащая использованию как имя ссылки на тип, пуста, ее следует заменить буквой "x" (ЛАТИНСКАЯ ПРОПИСНАЯ БУКВА X); и
- если строка символов, подлежащая использованию как идентификатор, пуста, ее следует заменить буквой "x" (ЛАТИНСКАЯ СТРОЧНАЯ БУКВА X).

10.3.4 В зависимости от вида генерируемого имени применяется один из трех следующих подразделов.

10.3.4.1 Если генерируемое имя – это имя ссылки на тип присвоения типов ASN.1, а строка символов, генерируемая согласно подразделу 10.3.3, идентична имени ссылки на тип другого присвоения типов ASN.1, ранее генерированного в том же модуле ASN.1 или в другом модуле ASN.1 с тем же пространством имен (включая отсутствие пространства имен), или является одним из резервируемых слов, описанных в Рекомендации МСЭ-Т X.680 | ИСО/МЭК 8824-1, 11.27, тогда к строке символов, генерируемой согласно подразделу 10.3.3, должен добавляться суффикс. Этот суффикс должен состоять из "ДЕФИСА-МИНУСА", за которым следует каноническое лексическое представление (см. часть 2, 2.3.1 Схемы языка XML консорциума W3C) целого числа. Это целое число должно быть таким наименьшим положительным числом, чтобы новое имя отличалось от имени ссылки на тип любого другого присвоения типов ASN.1, ранее генерированного в любом из этих модулей ASN.1.

10.3.4.2 Если генерируемое имя – это идентификатор компоненты последовательности, множества или типа выбора, а строка символов, генерируемая согласно подразделу 10.3.3, идентична идентификатору ранее генерированной компоненты той же последовательности, множества или типа выбора, тогда к строке символов, генерируемой согласно подразделу 10.3.3, должен прибавляться суффикс. Этот суффикс должен состоять из ДЕФИСА-МИНУСА, за которым следует каноническое лексическое представление (см. часть 2, 2.3.1 Схемы языка XML консорциума W3C) целого числа. Это целое число должно быть таким наименьшим положительным целым числом, чтобы новый идентификатор отличался от идентификатора любой ранее генерированной компоненты этой последовательности, множества или типа выбора.

10.3.4.3 Если генерируемое имя – это элемент "identifier" (идентификатор) в элементе "EnumerationItem" (элемент перечисления) перечислимого типа, а строка символов, генерируемая согласно подразделу 10.3.3, идентична элементу "identifier" в другом элементе "EnumerationItem", ранее генерированном в том же перечислимом типе, тогда к строке символов, генерируемой согласно подразделу 10.3.3, должен прибавляться суффикс. Этот суффикс должен состоять из ДЕФИСА-МИНУСА, за которым следует каноническое лексическое представление (см. часть 2, 2.3.1 Схемы языка XML консорциума W3C) целого числа. Это целое число должно быть таким наименьшим положительным целым числом, чтобы новый идентификатор отличался от элемента "identifier" в любом другом элементе "EnumerationItem", уже присутствующем в этом перечислимом типе ASN.1.

10.3.5 Что касается имени ссылки на тип (или идентификатора) ASN.1, которое генерируется в результате применения данного подраздела 10.3 к компоненте **name** компонент **element declaration**, **attribute declaration**, компоненты **complex type definition** верхнего уровня или определяемой пользователем компоненты **simple type definition** верхнего уровня, то если генерируемое имя ссылки на тип (или идентификатор) отличается от компоненты **name**, то конечная команда кодирования **NAME** будет присвоена присвоению типа ASN.1 с таким именем ссылки на тип (или компоненте с таким идентификатором), как описано в трех последующих подразделах.

10.3.5.1 Если единственное различие относится к регистру первой буквы (верхний регистр для имени ссылки на тип, а нижний регистр – для компоненты **name**), то элементом "Keyword" (ключевое слово) в команде кодирования **NAME** должно быть **UNCAPITALIZED**.

10.3.5.2 Если единственное различие относится к регистру первой буквы (нижний регистр для идентификатора, а верхний регистр – для компоненты **name**), то элементом "Keyword" (ключевое слово) в команде кодирования **NAME** должно быть **CAPITALIZED**.

10.3.5.3 В противном случае элементом "NewName" (новое имя) в команде кодирования **NAME** будет компонента **name**.

ПРИМЕР. – Компонента **complex type definition** на верхнем уровне:

```
<xsd:complexType name="COMPONENTS">
  <xsd:sequence>
    <xsd:element name="Elem" type="xsd:boolean"/>
    <xsd:element name="elem" type="xsd:integer"/>
    <xsd:element name="Elem-1" type="xsd:boolean"/>
    <xsd:element name="elem-1" type="xsd:integer"/>
  </xsd:sequence>
</xsd:complexType>
```

отображается в присвоении типа ASN.1:

```
COMPONENTS-1 ::= [NAME AS "COMPONENTS"] SEQUENCE {
  elem      [NAME AS CAPITALIZED] BOOLEAN,
  elem-1    [NAME AS "elem"] INTEGER,
  elem-1-1  [NAME AS "Elem-1"] BOOLEAN,
  elem-1-2  [NAME AS "elem-1"] INTEGER }
```

10.3.6 Что касается имени ссылки на тип (или идентификатора) ASN.1, которое генерируется в результате применения данного подраздела 10.3 к компоненте **name** компонент **element declaration**, **attribute declaration**, компоненты **complex type definition** верхнего уровня или определяемой пользователем компоненты **simple type definition** верхнего уровня, то если компонента **target namespace** схемной компоненты не обладает свойством **absent**, то конечная команда кодирования **NAMESPACE** должна быть присвоена присвоению типа ASN.1 с этим именем ссылки на тип (или именованному типу с этим идентификатором) и должна определять компоненту **target namespace** схемной компоненты.

10.3.7 Что касается идентификатора ASN.1, который генерируется в результате применения данного подраздела 10.3 для отображения компоненты **simple type definition** с аспектом **enumeration**, где генерируемый идентификатор отличается от соответствующего члена компоненты **value** аспекта **enumeration**, то конечная команда кодирования **TEXT** будет присвоена перечислимому типу ASN.1 с уточняющей информацией, определяющей элемент "identifier" в элементе "EnumerationItem" перечислимого типа. Применяется один из следующих двух подразделов.

10.3.7.1 Если единственное различие относится к регистру первой буквы (нижний регистр для идентификатора, а верхний регистр – для члена компоненты **value** сегмента **enumeration**), тогда элементом "Keyword" (ключевое слово) в команде кодирования **TEXT** будет **CAPITALIZED**.

10.3.7.2 В противном случае элемент "NewName" (Новое имя) в команде кодирования **TEXT** будет членом компоненты **value** аспекта **enumeration**.

10.4 Порядок отображения

10.4.1 Схемные компоненты верхнего уровня схемы XSD исходного текста должны подчиняться порядку выполнения отображения. Это относится к компонентам **model group definitions**, компонентам **complex type definition** верхнего уровня, определяемым пользователем компонентам **simple type definition** верхнего уровня, компонентам **attribute declaration** верхнего уровня и компонентам **element declaration** верхнего уровня.

ПРИМЕЧАНИЕ. – Прочие схемные компоненты верхнего уровня не отображаются в ASN.1, а встроенные типы данных XSD отображаются особым образом.

10.4.2 Этот порядок определяется в трех следующих подразделах.

10.4.2.1 Схемные компоненты верхнего уровня сначала должны упорядочиваться по их компоненте **target namespace**, с пространством имен со свойством **absent**, предшествующим всем именам пространства имен в возрастающем лексикографическом порядке.

10.4.2.2 В каждом целевом пространстве имен схемные компоненты верхнего уровня следует делить на четыре множества, упорядоченных следующим образом:

- a) **element declarations**;
- b) **attribute declarations**;
- c) **complex type definitions** и **simple type definitions**;
- d) **model group definitions**.

10.4.2.3 В пределах каждого множества (см. подраздел 10.4.2.2) схемные компоненты следует упорядочивать по компоненте **name** в возрастающем лексикографическом порядке.

10.4.3 При отображении генерируется ряд присвоений типа ASN.1, которые непосредственно не соответствуют любой схемной компоненте XSD, а именно:

- a) типы выбора (с конечной командой кодирования **USE-TYPE**), соответствующие иерархии получения типа; имена ссылок на типы для этих типов имеют суффикс **"-derivations"** (см. раздел 29);
- b) типы выбора (с конечной командой кодирования **USE-TYPE** по типу и конечной командой кодирования **USE-NIL** по каждой альтернативе), соответствующие иерархии получения типов, где определяемая пользователем компонента **simple type definition** или **complex type definition** на верхнем уровне, то есть корень иерархии получения типов, используется как компонента **type definition** одной или нескольких компонент **element declarations** со свойством **nillable**; имена ссылок на типы для этих типов имеют суффикс **"-deriv-nillable"** (см. раздел 29);
- c) типы выбора (с конечной командой кодирования **USE-TYPE** по типу и конечной командой кодирования **DEFAULT-FOR-EMPTY** по каждой альтернативе), соответствующие иерархии получения типов, где определяемая пользователем компонента **simple type definition** или **complex type definition** на верхнем уровне, то есть корень иерархии получения типов, используется как компонента **type definition** одной или нескольких компонент **element declaration** без свойства **nillable** и имеющих компоненту **value constraint**, которая является значением со свойством **default**; имена ссылок на типы для этих типов имеют суффикс **"-deriv-default-"** (см. раздел 29);
- d) типы выбора (с конечной командой кодирования **USE-TYPE** по типу и конечной командой кодирования **DEFAULT-FOR-EMPTY** по каждой альтернативе), соответствующие иерархии получения типов, где определяемая пользователем компонента **simple type definition** или **complex type definition** на верхнем уровне, то есть корень иерархии получения типов, используется как компонента **type definition** одной или нескольких компонент **element declaration**, которые не имеют свойства **nillable** и имеют компоненту **value constraint**, являющуюся значением со свойством **fixed**; имена ссылок на типы для этих типов имеют суффикс **"-deriv-fixed-"** (см. раздел 29);
- e) типы выбора (с конечной командой кодирования **USE-TYPE** по типу и конечными командами кодирования **USE-NIL** и **DEFAULT-FOR-EMPTY** по каждой альтернативе), соответствующие иерархии получения типов, где определяемая пользователем компонента **simple type definition** или **complex type definition** верхнего уровня, то есть корень иерархии получения типов, используется как компонента **type definition** одной или нескольких компонент **element declaration**, которые обладают свойством **nillable** и имеют компоненту **value constraint**, являющуюся значением со свойством **default**; имена ссылок на типы для этих типов имеют суффикс **"-deriv-nillable-default-"** (см. раздел 29);
- f) типы выбора (с конечной командой **USE-TYPE** по типу и конечными командами кодирования **USE-NIL** и **DEFAULT-FOR-EMPTY** по каждой альтернативе), соответствующие иерархии получения типов, где определяемая пользователем компонента **simple type definition** или **complex type definition** верхнего уровня, то есть корень иерархии получения типов, используется как компонента **type definition** одной или нескольких компонент **element declaration**, обладающих свойством **nillable** и имеющих компоненту **value constraint**, которая является значением со свойством **fixed**; имена ссылок на типы для этих типов имеют суффикс **"-deriv-nillable-fixed-"** (см. раздел 29);

- g) типы выбора (с конечной командой кодирования **UNTAGGED**), соответствующие группе подстановки элементов, имена ссылки на типы для этих типов имеют суффикс **"-group"** (см. раздел 31);
- h) типы последовательности (с конечной командой кодирования **USE-NIL**), соответствующие использованию определяемых пользователем компонент **simple type definition** или **complex type definition** верхнего уровня как компонента **type definition** одной или нескольких компонент **element declaration**, обладающих свойством **nillable**; имена ссылок на типы для этих типов имеют суффикс **"-nillable"** (см. раздел 30).

10.4.4 Все присвоения типов ASN.1, которые непосредственно соответствуют схемным компонентам XSD в схеме XSD исходного текста, должны генерироваться перед всеми присвоениями типов ASN.1, перечисленными в подразделе 10.4.3 (если таковые имеются).

10.4.5 Присвоения типов ASN.1, непосредственно соответствующих схемным компонентам XSD, должны генерироваться согласно порядку соответствующих схемных компонент XSD (см. подраздел 10.4.1). Присвоения типов ASN.1, перечисленные в подразделе 10.4.3 (если таковые имеются), должны генерироваться согласно порядку схемных компонент XSD (см. подраздел 10.4.1), соответствующих "ассоциированному присвоению типа" (см. разделы 29, 30 и 31).

10.4.6 Что касается подраздела 10.4.3, пунктов от c) до f), то если компонента **simple type definition** или **complex type definition**, то есть корень иерархии получения типов, используется как компонента **type definition** множества компонент **element declaration**, имеющих разные значения в компоненте **value constraint**, то присвоения типов ASN.1 должны генерироваться в возрастающем лексикографическом порядке канонического лексического представления (см. часть 2, 2.3.1 Схемы языка XML консорциума W3C) значения в компоненте **value constraint**.

11 Отображение использования встроенных типов данных XSD

11.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего использованию встроенного типа данных XSD.

11.2 Использование встроенного типа данных XSD будет отображаться в определении типа ASN.1 в соответствии с таблицей 2. В этой таблице дано используемое определение типов ASN.1. Обозначение "XSD.Name" указывает на то, что определение типа ASN.1 должно быть определением типа ASN.1 ("DefinedType") (Определенный тип), генерируемым путем применения положений подраздела 10.2 к соответствующему присвоению типа ASN.1, имеющемуся в модуле XSD.

Таблица 2 – Определения типов ASN.1, соответствующих использованиям встроенных типов данных XSD

Встроенный тип данных XSD	Определение типа ASN.1	Встроенный тип данных XSD	Определение типа ASN.1
anyURI	XSD.AnyURI	int	XSD.Int
anySimpleType	XSD.AnySimpleType	integer	INTEGER
anyType	XSD.AnyType	language	XSD.Language
base64Binary	[BASE64] OCTET STRING	long	XSD.Long
boolean	BOOLEAN	Name	XSD.Name
byte	INTEGER (-128..127)	NCName	XSD.NCName
date	XSD.Date	negativeInteger	INTEGER (MIN..-1)
dateTime	XSD.DateTime	NMTOKEN	XSD.NMTOKEN
decimal	XSD.Decimal	NMTOKENS	XSD.NMTOKENS
double	XSD.Double	nonNegativeInteger	INTEGER (0..MAX)
duration	XSD.Duration	nonPositiveInteger	INTEGER (MIN..0)
ENTITIES	XSD.ENTITIES	normalizedString	XSD.NormalizedString
ENTITY	XSD.ENTITY	NOTATION	XSD.NOTATION
float	XSD.Float	positiveInteger	INTEGER (1..MAX)
gDay	XSD.GDay	QName	XSD.QName
gMonth	XSD.GMonth	short	XSD.Short
gMonthDay	XSD.GMonthDay	string	XSD.String
gYear	XSD.GYear	time	XSD.Time
gYearMonth	XSD.GYearMonth	token	XSD.Token
hexBinary	OCTET STRING	unsignedByte	INTEGER (0..255)
ID	XSD.ID	unsignedInt	XSD.UnsignedInt
IDREF	XSD.IDREF	unsignedLong	XSD.UnsignedLong
IDREFS	XSD.IDREFS	unsignedShort	XSD.UnsignedShort

12 Отображение аспектов

Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для отображения компоненты **facet** (аспект) компоненты **simple type definition**. STD для этого сочетания компонент отображается в ограничении ASN.1, применяемом к определению типа ASN.1, соответствующего STD, если STD не имеет аспекта **enumeration**, который отображается в элементе "Enumeration" (Перечисление) ASN.1 (см. подразделы 12.4.1 и 12.4.2). В этом случае ограничение ASN.1 не генерируется из аспекта (см. подразделы 12.1.2, 12.2.1, 12.3.1 и 12.5.1).

12.1 Аспекты length, minLength и maxLength

12.1.1 Аспекты **length**, **minLength** и **maxLength** должны игнорироваться для встроенных типов данных XSD **xsd:QName** и **xsd:NOTATION** и для любой компоненты **simple type definition**, полученной из них путем ограничения.

12.1.2 Если аспект **length**, **minLength** или **maxLength** принадлежит компоненте **simple type definition**, которая также имеет аспект **enumeration**, отображаемый в элементе "Enumeration" ASN.1 (см. подразделы 12.4.1 и 12.4.2), то элемент "EnumerationItem" не должен включаться в элемент "Enumeration" для членов (если таковые имеются) компоненты **value** аспекта **enumeration**, которые не удовлетворяют аспектам **length**, **minLength** или **maxLength**.

12.1.3 В противном случае аспекты **length**, **minLength** и **maxLength** компоненты **simple type definition** должны отображаться в ограничении размера ASN.1 согласно таблице 3.

Таблица 3 – Ограничения размеров ASN.1, соответствующие аспектам **length**, **minLength**, и **maxLength**

Аспект XSD	Ограничения размера ASN.1
length=значение	(SIZE(значение))
minLength=min	(SIZE(min .. MAX))
maxLength=max	(SIZE(0 .. max))
minLength=min maxLength=max	(SIZE(min .. max))

12.2 Аспект pattern

12.2.1 Если аспект **pattern** принадлежит компоненте **simple type definition**, которая также имеет аспект **enumeration**, отображаемый в элементе "Enumeration" ASN.1 (см. подразделы 12.4.1 и 12.4.2), то элемент "EnumerationItem" не следует включать в элемент "Enumeration" для членов (если таковые имеются) компоненты **value** аспекта **enumeration**, которые не удовлетворяют аспекту **pattern**.

12.2.2 В противном случае аспект **pattern** должен отображаться в ограничении, определяемом пользователем. Применяется один из двух следующих подразделов.

12.2.2.1 Если компонента **value** аспекта **pattern** является простым регулярным выражением, ограничением, определяемым пользователем, будет:

(CONSTRAINED BY { /* Представление XML комбинации "хуз" XCD */ }),

где "хуз" – это представление на языке XML компоненты **value** аспекта **pattern**, за исключением случая, когда в компоненте **value** аспекта **pattern** появляется подстрока "*" и она должна быть заменена строкой символов "/".

12.2.2.2 Если компонента **value** аспекта **pattern** есть конъюнкция объединений регулярных выражений (общий случай), то определяемое пользователем ограничение не задается (см. подраздел 12.5.4).

12.3 Аспект whiteSpace

12.3.1 Если аспект **whiteSpace** с компонентой **value** в компонентах **replace** или **collapse** принадлежит компоненте **simple type definition**, которая имеет также аспект **enumeration**, отображаемый в элементе "Enumeration" ASN.1 (см. подразделы 12.4.1 и 12.4.2), то элемент "EnumerationItem" не должен включаться в элемент "Enumeration" для членов (если таковые имеются) компоненты **value** аспекта **enumeration**, которые содержат любой из символов HORIZONTAL TABULATION (ГОРИЗОНТАЛЬНОЕ ТАБУЛИРОВАНИЕ), NEWLINE (НОВАЯ СТРОКА) или CARRIAGE RETURN (ВОЗВРАТ КАРЕТКИ), или (в случае компоненты **collapse**) содержат интерлиньяж, конецик или множество последовательных символов SPACE (ПРОБЕЛ).

12.3.2 В противном случае применяется максимум один из следующих подразделов:

12.3.2.1 Если компонента **value** аспекта **whiteSpace** имеет свойство **preserve**, этот аспект следует игнорировать.

12.3.2.2 Если компонента **value** аспекта **whiteSpace** имеет свойство **replace**, а определение типа ASN.1, соответствующее компоненте **simple type definition**, является типом ограниченной строки символов ASN.1, тогда следует добавить допустимое ограничение алфавита к определению типа ASN.1 для устранения символов

HORIZONTAL TABULATION, NEWLINE и CARRIAGE RETURN. Конечная команда кодирования **WHITESPACE REPLACE** должна быть присвоена определению типа ASN.1. Будет использовано следующее или эквивалентное допустимое алфавитное ограничение:

```
(FROM ({0, 0, 0, 32} .. {0, 16, 255, 255}))
```

12.3.2.3 Если компонента **value** аспекта **whiteSpace** имеет свойство **collapse**, а определение типа ASN.1, соответствующее компоненте **simple type definition**, является ограниченным типом строки символов, то как допустимое алфавитное ограничение, упомянутое в подразделе 12.3.2.2, так и ограничение кодовой комбинации, запрещающее интерлиньяж, концевик и множество последовательных символов SPACE, следует добавить к определению типа ASN.1. Конечная команда кодирования **WHITESPACE COLLAPSE** должна присваиваться определению типа ASN.1. Следует использовать следующее или эквивалентное ограничение кодовой последовательности:

```
(PATTERN "([^\ ]([^\ ]| [^\ ]*))?")
```

12.4 Аспект enumeration

12.4.1 Аспект **enumeration**, принадлежащий компоненте **simple type definition** с компонентой **variety** свойства **atomic**, полученный путем ограничения (прямо или косвенно) из определения типа ASN.1 **xsd:string**, не должен отображаться в ограничении ASN.1. Вместо этого данный аспект следует отображать в элементе "Enumeration" перечислимого типа ASN.1, соответствующего компоненте **simple type definition** (см. подраздел 13.5), как определено в следующих трех подразделах.

12.4.1.1 Для каждого члена компоненты **value** аспекта **enumeration** элемент "EnumerationItem", являющийся элементом "identifier" ("идентификатор"), следует прибавлять к элементу "Enumeration" (согласно подразделам 12.1.2, 12.2.1, 12.3.1 и 12.5.1).

12.4.1.2 Каждый элемент "identifier" должен генерироваться путем применения положений подраздела 10.3 к соответствующему члену компоненты **value** аспекта **enumeration**.

12.4.1.3 Члены компоненты **value** аспекта **enumeration** должны отображаться в возрастающем лексикографическом порядке, а все дублируемые члены следует отбрасывать.

12.4.2 Аспект **enumeration**, принадлежащий компоненте **simple type definition** с компонентой **variety** свойства **atomic**, полученный путем ограничения (прямо или косвенно) из определения типа ASN.1 **xsd:integer**, не должен отображаться в ограничении ASN.1. Вместо этого данный аспект следует отображать в элементе "Enumeration" перечислимого типа ASN.1, соответствующего компоненте **simple type definition** (см. подраздел 13.6), как определено в следующих трех подразделах.

12.4.2.1 Для каждого члена компоненты **value** аспекта **enumeration** элемент "EnumerationItem", являющийся элементом "NamedNumber" ("Именованное число") будет добавляться к элементу "Enumeration" (см. подразделы 12.1.2, 12.2.1, 12.3.1 и 12.5.1).

12.4.2.2 Элемент "identifier" в каждом элементе "NamedNumber" будет генерироваться путем конкатенации строки символов "int" с каноническим лексическим представлением (см. часть 2, 2.3.1 Схемы языка XML консорциума W3C) соответствующего члена компоненты **value** аспекта **enumeration**. Элемент "SignedNumber" ("Число со знаком") в элементе "NamedNumber" ("Именованное число") будет обозначением значения в ASN.1 для члена (целое число).

12.4.2.3 Члены компоненты **value** аспекта **enumeration** должны отображаться в возрастающем числовом порядке, а все дублируемые члены следует отбрасывать.

12.4.3 Любой другой аспект **enumeration** должен отображаться в ограничении ASN.1, которое является либо единичным значением, либо объединением единичных значений, соответствующих членам компоненты **value** аспекта **enumeration**.

ПРИМЕЧАНИЕ. – Аспект **enumeration** применяется к пространству значений компоненты **base type definition**. Поэтому для аспекта **enumeration** встроенных типов данных XSD **xsd:QName** или **xsd:NOTATION** значение компоненты **uri** в компоненте **[USE-QNAME] SEQUENCE**, полученной как ограничение ASN.1 единичного значения, определяется в представлении в языке XML схемы XSD путем описаний пространства имен, область определения которых включает типы данных **xsd:QName** или **xsd:NOTATION**, и с помощью префикса (если таковой имеется) типов данных **xsd:QName** или **xsd:NOTATION**.

ПРИМЕР 1. – Ниже приводится определяемая пользователем компонента **simple type definition** верхнего уровня, являющаяся ограничением определения типа **xsd:string** с аспектом **enumeration**:

```
<xsd:simpleType name="state">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="off"/>
    <xsd:enumeration value="on"/>
  </xsd:restriction>
</xsd:simpleType>
```

Она отображается в присвоении типа ASN.1:

```
State ::= [NAME AS UNCAPITALIZED] ENUMERATED {off, on}
```

ПРИМЕР 2. – Ниже представлена определяемая пользователем компонента **simple type definition** верхнего уровня, которая является ограничением определения типа **xsd:integer** с аспектом **enumeration**:

```
<xsd:simpleType name="integer-0-5-10">
  <xsd:restriction base="xsd:integer">
    <xsd:enumeration value="0"/>
    <xsd:enumeration value="5"/>
    <xsd:enumeration value="10"/>
  </xsd:restriction>
</xsd:simpleType>
```

Она отображается в присвоении типа ASN.1:

```
Integer-0-5-10 ::= [NAME AS UNCAPITALIZED] ENUMERATED {int0(0), int5(5), int10(10)}
```

ПРИМЕР 3. – Ниже представлена определяемая пользователем компонента **simple type definition** верхнего уровня, которая является ограничением определения типа **xsd:integer** с аспектами **minInclusive** и **maxInclusive**:

```
<xsd:simpleType name="integer-1-10">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="10"/>
  </xsd:restriction>
</xsd:simpleType>
```

Она отображается в присвоении типа ASN.1:

```
Integer-1-10 ::= [NAME AS UNCAPITALIZED] INTEGER(1..10)
```

ПРИМЕР 4. – Ниже представлена определяемая пользователем компонента **simple type definition** верхнего уровня, которая является ограничением (с аспектом **minExclusive**) другой компоненты **simple type definition**, полученной путем ограничения из определения типа **xsd:integer** с добавлением аспектов **minInclusive** и **maxInclusive**:

```
<xsd:simpleType name="multiple-of-4">
  <xsd:restriction>
    <xsd:simpleType>
      <xsd:restriction base="xsd:integer">
        <xsd:minInclusive value="1"/>
        <xsd:maxInclusive value="10"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:minExclusive value="5"/>
  </xsd:restriction>
</xsd:simpleType>
```

Она отображается в присвоении типа ASN.1:

```
Multiple-of-4 ::= [NAME AS UNCAPITALIZED] INTEGER(5<..10)
```

ПРИМЕР 5. – Ниже представлена определяемая пользователем компонента **simple type definition** верхнего уровня, которая является ограничением (с аспектами **minLength** и **maxLength**) другой компоненты **simple type definition**, полученной путем ограничения из определения типа **xsd:string** с добавлением аспекта **enumeration**:

```
<xsd:simpleType name="color">
  <xsd:restriction>
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="white"/>
        <xsd:enumeration value="black"/>
        <xsd:enumeration value="red"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:minLength value="2"/>
    <xsd:maxLength value="4"/>
  </xsd:restriction>
</xsd:simpleType>
```

Она отображается в присвоении типа ASN.1:

```
Color ::= [NAME AS UNCAPITALIZED] ENUMERATED {red}
```

12.5 Прочие аспекты

12.5.1 Если аспект **totalDigits**, **fractionDigits**, **maxInclusive**, **maxExclusive**, **minExclusive** или **minInclusive** принадлежит компоненте **simple type definition**, которая также имеет аспект **enumeration**, отображаемый в элементе "Enumeration" ASN.1 (см. подразделы 12.4.1 и 12.4.2), то элементы "EnumerationItem" не будут включаться в элемент "Enumeration" для членов (если таковые имеются) компоненты **value** с аспектом **enumeration**, которые не удовлетворяют аспекту **totalDigits**, **fractionDigits**, **maxInclusive**, **maxExclusive**, **minExclusive** или **minInclusive**.

12.5.2 Если аспект **maxInclusive**, **maxExclusive**, **minExclusive** или **minInclusive** принадлежит компоненте **simple type definition** без аспекта **enumeration** или с аспектом **enumeration**, который не отображается в элементе "Enumeration" ASN.1 (см. подразделы 12.4.1 и 12.4.2), применяется один из двух следующих подразделов:

12.5.2.1 Если компонента **simple type definition** получена путем ограничения (непосредственно или косвенно) из встроенного типа данных "дата или время" (**xsd:date**, **xsd:dateTime**, **xsd:duration**, **xsd:gDay**, **xsd:gMonth**, **xsd:gYear**, **xsd:gYearMonth**, **xsd:gMonthDay** или **xsd:time**), то аспекты **maxInclusive**, **maxExclusive**, **minExclusive** и **minInclusive** компоненты **simple type definition** должны отображаться в определяемом пользователем ограничении ASN.1 (см. подраздел 12.5.4).

12.5.2.2 В противном случае аспекты **maxInclusive**, **maxExclusive**, **minExclusive** и **minInclusive** компоненты **simple type definition** должны отображаться в ограничении диапазона значений или ограничении единичного значения ASN.1 согласно таблице 4.

Таблица 4 – Ограничения ASN.1, соответствующие аспектам **maxInclusive, **maxExclusive**, **minExclusive** и **minInclusive****

Аспект XSD	Ограничение ASN.1
maxInclusive=ub	(MIN .. ub)
maxExclusive=ub	(MIN .. < ub)
minExclusive=lb	(lb < .. MAX)
minInclusive=lb	(lb .. MAX)
minInclusive=ub maxInclusive=lb	(lb .. ub)
minInclusive=v maxInclusive=v	(v)
minInclusive=ub maxExclusive=lb	(lb .. < ub)
minExclusive=ub maxInclusive=lb	(lb < .. ub)
minExclusive=ub maxExclusive=lb	(lb < .. < ub)

12.5.3 Если аспект **totalDigits** или **fractionDigits** принадлежит компоненте **simple type definition** без аспекта **enumeration** или с аспектом **enumeration**, который не отражается в элементе "Enumeration" ASN.1 (см. подразделы 12.4.1 и 12.4.2), аспекты **totalDigits** и **fractionDigits** компоненты **simple type definition** должны отображаться в определяемом пользователем ограничении (см. подраздел 12.5.4).

12.5.4 Когда тот или иной аспект отображается в определяемом пользователем ограничении ASN.1, рекомендуется, чтобы аспект и его компонента **value** присутствовали в замечании ASN.1 в определяемом пользователем ограничении. Точная форма определяемого пользователем ограничения не установлена.

13 Отображение компонент **simple type definition** (определение простого типа)

13.1 Этот раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования присвоения типа ASN.1 или определения типа ASN.1, соответствующих компоненте **simple type definition** (определение простого типа).

13.2 В данном разделе описывается отображение компонент **simple type definition**, которые не являются встроенными типами данных XSD. Множество встроенных типов данных XSD отображается в предопределенном модуле ASN.1, описанном в Приложении А (модуль **xsd**), который следует включать в спецификации ASN.1, генерируемые отображением.

13.3 Определяемая пользователем компонента **simple type definition** верхнего уровня должна отображаться в присвоении типа ASN.1. Элемент "typereference" (ссылка на тип) в элементе "TypeAssignment" (присвоение типа) должен генерироваться путем применения положений подраздела 10.3 к компоненте **name** в компоненте **simple type definition**, а элемент "Type" (Тип) в элементе "TypeAssignment" должен быть определением типа ASN.1, как описано в подразделах 13.5–13.10.

13.4 Анонимная компонента **simple type definition** должна отображаться в определении типа ASN.1, как описано в подразделах 13.5–13.10.

13.5 Что касается компоненты **simple type definition** с компонентой **variety** свойства **atomic** с аспектом **enumeration**, получаемой путем ограничения (прямо или косвенно) из определения типа **xsd:string**, то определение типа ASN.1 должно быть перечислимым типом ASN.1, чей элемент "Enumeration" должен генерироваться, как описано в подразделе 12.4.1.

13.6 Что касается компоненты **simple type definition** с компонентой **variety** свойства **atomic** с аспектом **enumeration**, получаемой путем ограничения (прямо или косвенно) из определения типа **xsd:integer**, то определение типа ASN.1 должно быть перечислимым типом ASN.1, чей элемент "Enumeration" должен генерироваться, как описано в подразделе 12.4.2. Конечная команда кодирования **USE-NUMBER** должна присваиваться перечислимому типу ASN.1.

13.7 Что касается любой другой компоненты **simple type definition** (например, D) с любой компонентой **variety**, полученной путем ограничения (прямо или косвенно) из определяемой пользователем компоненты **simple type definition** верхнего уровня, то определение типа ASN.1 будет генерироваться путем применения раздела 23 к определяемой пользователем компоненте **simple type definition** верхнего уровня (например, B), такой, что:

- a) D получается из B путем ограничения (прямо или косвенно); и
- b) либо B является компонентой **base type definition** применительно к D, либо все промежуточные шаги перехода от B к D являются анонимными компонентами **simple type definitions**.

Тогда для каждого из аспектов (**facets**) определения D (если таковые имеются) ограничение ASN.1, генерируемое путем применения раздела 12 к аспекту, должно прибавляться к определению типа ASN.1.

13.8 Что касается любой другой компоненты **simple type definition** (например, D) с компонентой **variety** свойства **atomic**, то определение типа ASN.1 должно генерироваться путем применения раздела 23 к встроенному типу данных XSD (например, B) так, что:

- a) D получается из B путем ограничения (прямо или косвенно); и
- b) либо B является компонентой **base type definition** для D, либо все промежуточные шаги перехода от B к D являются анонимными компонентами **simple type definitions**.

Тогда для каждого из аспектов (**facets**) определения D ограничение ASN.1, генерируемое путем применения раздела 12 к аспекту, будет добавляться к определению типа ASN.1.

13.9 Что касается любой другой компоненты **simple type definition** (например, D) с компонентой **variety** в компоненте **list**, то применяются три следующих подраздела.

13.9.1 Определением типа ASN.1 будет последовательный тип ASN.1, компонента которого будет элементом "Type", генерируемым в результате применения раздела 12 к компоненте **item type definition**.

13.9.2 Для каждого из аспектов (**facets**) определения типа D ограничение ASN.1, генерируемое путем применения раздела 12 к аспекту, будет добавляться к типу последовательности ASN.1.

13.9.3 Конечная команда кодирования **LIST** будет присваиваться типу последовательности ASN.1.

ПРИМЕР. – Ниже представлена определяемая пользователем компонента **simple type definition** верхнего уровня, являющаяся компонентом **list** определения типа **xsd:float**:

```
<xsd:simpleType name="list-of-float">
  <xsd:list itemType="xsd:float"/>
</xsd:simpleType>
```

Она отображается в присвоении типа ASN.1:

```
List-of-float ::= [LIST] [NAME AS UNCAPITALIZED] SEQUENCE OF XSD.Float
```

13.10 Для любой другой компоненты **simple type definition** (например, D) с компонентой **variety** в компоненте **union** применяются пять следующих подразделов.

13.10.1 Определение типа ASN.1 должно быть типом выбора ASN.1 с одной альтернативой для каждого члена компоненты **member type definitions** (определения типов членов).

13.10.2 Для каждого члена компоненты **member type definitions** элемент "identifier" в элементе "NamedType" соответствующей альтернативы должен генерироваться путем применения подраздела 10.3 к компоненте **name** члена (если член является встроенным типом данных XSD или определяемой пользователем компонентой **simple type definition** верхнего уровня) или к строке символов "alt" (если член является анонимной компонентой **simple type definition**), а элемент "Type" в элементе "NamedType" должен быть определением типа ASN.1, генерируемым путем применения раздела 23 к члену компоненты **member type definitions**.

13.10.3 Для каждого члена компоненты **member type definitions**, являющейся анонимной компонентой **simple type definition**, соответствующий элемент "NamedType" будет иметь конечную команду кодирования **NAME AS ""**.

13.10.4 Для каждого из аспектов (**facets**) определения типа D ограничение ASN.1, генерируемое путем применения раздела 12 к аспекту, должно добавляться к типу выбора ASN.1.

13.10.5 Конечная команда кодирования **USE-UNION** должна присваиваться типу выбора ASN.1.

ПРИМЕР. – Ниже представлена определяемая пользователем компонента **simple type definition** верхнего уровня, являющаяся компонентой **union** (объединение) двух анонимных компонент **simple type definitions**:

```
<xsd:simpleType name="decimalOrBinary">
  <xsd:union>
    <xsd:simpleType>
      <xsd:restriction base="xsd:decimal"/>
    </xsd:simpleType>
    <xsd:simpleType>
      <xsd:restriction base="xsd:float"/>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
```

Она отображается в присвоении типа ASN.1:

```
DecimalOrBinary ::= [NAME AS UNCAPITALIZED] [USE-UNION] CHOICE {
    alt          [NAME AS ""] XSD.Decimal,
    alt-1        [NAME AS ""] XSD.Float }
```

14 Отображение компонент **element declaration** (описание элемента)

14.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования присвоения типов ASN.1 или определения типов ASN.1, соответствующих компоненте **element declaration** (описание элемента).

ПРИМЕЧАНИЕ. – Наличие компоненты **value constraint** (ограничение значения) в компоненте **element declaration** обычно влияет на отображение. Вместе с тем подразделом 8.10 подразумевается, что компонента **element declaration**, имеющая компоненту **value constraint** и у которой компонента **type definition** является определением типа **xsd:Qname**, **xsd:NOTATION** или ограничением этих встроенных типов данных XSD, отображается так, как если бы она не имела компоненты **value constraint**.

14.2 Компонента **element declaration** верхнего уровня со свойством **abstract** должна игнорироваться.

14.3 Компонента **element declaration** верхнего уровня без свойства **abstract** должна отображаться в присвоении типа ASN.1. Элемент "typereference" (ссылка на тип) в элементе "TypeAssignment" должен генерироваться путем применения подраздела 10.3 к компоненте **name** для компоненты **element declaration**, а элемент "Type" в элементе "TypeAssignment" должен быть определением типа ASN.1, как описано в подразделе 14.5.

14.4 Локальная компонента **element declaration** должна отображаться в определении типа ASN.1, как описано в подразделе 14.5.

14.5 Применяется один из двух следующих подразделов (подразделы 14.5.1 и 14.5.2).

14.5.1 Если компонента **type definition** в компоненте **element declaration** является анонимной компонентой **simple type definition**, **complex type definition** или встроенным типом данных XSD (например, A), то применяется один из двух следующих подразделов.

14.5.1.1 Если компонента **element declaration** не обладает свойством **nillable**, то определение типа ASN.1 должно генерироваться путем применения раздела 23 к типу A.

14.5.1.2 Если компонента **element declaration** имеет свойство **nillable**, то определение типа ASN.1 должно генерироваться путем применения к типу A раздела 26 (если A – компонента **simple type definition**) либо раздела 27 (если A – компонента **complex type definition**).

14.5.2 Если компонента **type definition** компоненты **element declaration** является определяемой пользователем компонентой **simple type definition** или компонентой **complex type definition** (например, T) на верхнем уровне, то применяется один из следующих четырех подразделов.

14.5.2.1 Если компонента **element declaration** не имеет свойства **nillable** и не имеет заменяемой компоненты **type definition** (см. подраздел 14.6), то определение типа ASN.1 должно генерироваться в результате применения раздела 23 к определению типа T.

14.5.2.2 Если компонента **element declaration** обладает свойством **nillable** и не имеет заменяемой компоненты **type definition** (см. подраздел 14.6), то определение типа ASN.1 должно генерироваться путем применения к определению типа T раздела 26 (если T – компонента **simple type definition**) или раздела 27 (если T – компонента **complex type definition**).

14.5.2.3 Если компонента **element declaration** не обладает свойством **nillable** и имеет заменяемую компоненту **type definition** (см. подраздел 14.6), то определение типа ASN.1 должно генерироваться путем применения раздела 24 к типу T.

14.5.2.4 Если компонента **element declaration** обладает свойством **nillable** и имеет заменяемую компоненту **type definition** (см. подраздел 14.6), то определение типа ASN.1 должно генерироваться путем применения раздела 25 к типу T.

14.6 Фраза "имеет заменяемую компоненту **type definition**", применяемая к компоненте **element declaration**, означает, что компонента **type definition** в компоненте **element declaration** является определяемой пользователем компонентой **simple type definition** или **complex type definition** верхнего уровня, которая возникает как компонента **base type definition** другой компоненты **simple type definition** или **complex type definition** верхнего уровня.

15 Отображение компонент **attribute declaration** (описание атрибута)

15.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования присвоения типа ASN.1 или определения типа ASN.1, соответствующих компоненте **attribute declaration** (описание атрибута).

15.2 Компонента **attribute declaration** верхнего уровня должна отображаться в присвоении типа ASN.1. Элемент "typereference" (ссылка на тип) в элементе "TypeAssignment" (присвоение типа) должен генерироваться в результате применения подраздела 10.3 к компоненте **name** компоненты **attribute declaration**, а элемент "Type" в элементе "TypeAssignment" должен быть определением типа ASN.1, как описано в подразделе 15.4. Конечная команда кодирования **ATTRIBUTE** должна присваиваться присвоению типа ASN.1.

15.3 Локальная компонента **attribute declaration** должна отображаться в определении типа ASN.1, как описано в подразделе 15.4.

15.4 Определение типа ASN.1 должно генерироваться в результате применения раздела 23 к компоненте **type definition** в компоненте **attribute declaration**.

16 Отображение значений компонент **simple type definition** (определение простого типа)

16.1 Этот раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования элемента "Value" (значение) ASN.1, соответствующего значению в пространстве значений компоненты **simple type definition**.

16.2 При заданном значении V в пространстве значений компоненты **simple type definition**, и:

- при определении типа ASN.1, отображенного из этой компоненты **simple type definition**; и
- при каноническом лексическом представлении (см. часть 2, 2.3.1 Схемы языка XML консорциума W3C) значения V,

V должно отображаться в базовое обозначение значения ASN.1 для абстрактного значения определения типа ASN.1, для которого в расширенных правилах XER (EXTENDED-XER) каноническое лексическое представление является действительным кодированием элемента "ExtendedXMLValue" (значение в расширенном языке XML).

17 Отображение компонент **model group definition** (определение группы моделей)

17.1 Этот раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования присвоения типа ASN.1, соответствующего компоненте **model group definition** (определение группы моделей).

17.2 Компонента **model group definition**, компонента которой **model group** (группа моделей) имеет компоненту **compositor** (составитель) для компонент **sequence** (последовательность) или **choice** (выбор), должна отображаться в присвоении типа ASN.1. Элемент "typereference" в элементе "TypeAssignment" должен генерироваться в результате применения подраздела 10.3 к компоненте **name** компоненты **model group definition**, а элемент "Type" в элементе "TypeAssignment" должен генерироваться в результате применения раздела 18 к компоненте **model group** в компоненте **model group definition**.

ПРИМЕЧАНИЕ. – Компоненты **model group definitions**, компонента которых **model group** имеет компоненту **compositor** со свойством **all**, не отображаются в ASN.1.

18 Отображение компонент **model group** (группа моделей)

18.1 Этот раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего компоненте **model group**.

ПРИМЕЧАНИЕ. – Этот раздел не применяется к каждой компоненте **model group**. Например, компонента **model group** с компонентой **compositor** со свойством **all** не отображается в ASN.1, но ее компоненты **particle** (частица) отображаются так, как описано в подразделе 20.9.

18.2 Компонента **model group** с компонентом **compositor** для компоненты **sequence** должна отображаться в последовательном типе ASN.1. Для каждой компоненты **particle** в компоненте **model group** по порядку элемент "NamedType" должен генерироваться в результате применения раздела 19 к компоненте **particle**, и этот элемент "NamedType" должен добавляться к последовательному типу как одна из его компонент. Конечная команда кодирования **UNTAGGED** должна присваиваться последовательному типу.

18.3 Компонента **model group** с компонентой **compositor** для компоненты **choice** должна отображаться в типе выбора ASN.1. Для каждой компоненты **particle** в компоненте **model group** по порядку элемент "NamedType" должен генерироваться в результате применения раздела 19 к компоненте **particle**, и этот элемент "NamedType" должен добавляться к типу выбора как одна из его альтернатив. Конечная команда кодирования **UNTAGGED** будет присваиваться типу выбора.

19 Отображение компонент **particle** (частица)

19.1 Этот раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования элемента "NamedType" ASN.1, соответствующего компоненте **particle**.

ПРИМЕЧАНИЕ. – Этот раздел не применяется ко всем компонентам **particle**. Например, (самая важная) компонента **particle** компоненты **content type** (содержательный тип) в компоненте **complex type definition** отображается особым образом, если ее компонента **term** является компонентой **model group** с компонентой **compositor** в компонентах **sequence** или **all** (см. подраздел 20.8).

19.2 Три следующих подраздела определяют термины, которые используются в остальной части раздела 19.

19.2.1 Если компоненты **min occurs** и **max occurs** в компоненте **particle** являются единицами, то компонента **particle** называется "частицей обязательного присутствия".

19.2.2 Если компонента **min occurs** является нулем, а компонента **max occurs** является единицей, то:

- если отображение компоненты **particle** должно генерировать компоненту последовательного типа ASN.1, то компонента **particle** называется "частицей факультативного присутствия";
- в противном случае компонента **particle** называется "факультативной частицей единичного вхождения".

19.2.3 Если значение компоненты **max occurs** составляет два или больше, компонента **particle** называется "частицей множественного вхождения".

19.3 "Частица обязательного присутствия" или "частица факультативного присутствия" должны отображаться в элементе "NamedType", как описано в двух следующих подразделах.

19.3.1 Элемент "identifier" в элементе "NamedType" должен генерироваться в результате применения подраздела 10.3 к строке символов, описанной в подразделе 19.5, а элемент "Type" в элементе "NamedType" будет генерироваться в результате применения подраздела 19.6 к компоненте **term** в компоненте **particle**.

19.3.2 Если компонента **particle** является "частицей факультативного присутствия", то элемент "NamedType" должен сопровождаться ключевым словом **OPTIONAL** (факультативный).

19.4 "Факультативная частица единичного вхождения" или "частица множественного вхождения" должна отображаться в элементе "NamedType", как описано в шести следующих подразделах.

19.4.1 Элемент "identifier" в элементе "NamedType" должен генерироваться в результате применения подраздела 10.3 к строке символов, полученной в результате добавления суффикса **"-list"** к строке символов, указанной в подразделе 19.5. Элемент "Type" в элементе "NamedType" должен быть типом последовательности.

19.4.2 Если компонента **particle** является "факультативной частицей единичного вхождения" или "частицей множественного вхождения", то к типу последовательности должно добавляться ограничение размера согласно таблице 5.

Таблица 5 – Ограничение размера ASN.1, соответствующее компонентам min occurs и max occurs

Компоненты min occurs и max occurs	Ограничение размера ASN.1
min occurs = <i>n</i> max occurs = <i>n</i> <i>n</i> ≥ 2	SIZE (<i>n</i>)
min occurs = <i>min</i> max occurs = <i>max</i> <i>max</i> > <i>min</i> and <i>max</i> ≥ 2	SIZE (<i>min</i> .. <i>max</i>)
min occurs = 0 max occurs = 1	SIZE (0 .. 1)
min occurs = <i>min</i> max occurs = unbounded <i>min</i> ≥ 1	SIZE (<i>min</i> .. MAX)
min occurs = 0 max occurs = unbounded	ограничения размера нет

19.4.3 Если компонента **term** компоненты **particle** является компонентой **element declaration**, то компонента типа последовательности будет элементом "NamedType". Элемент "identifier" в этом элементе "NamedType" будет генерироваться в результате применения подраздела 10.3 к компоненте **name** в компоненте **element declaration**, а элемент "Type" в этом элементе "NamedType" будет генерироваться в результате применения подраздела 19.6 к компоненте **term** в компоненте **particle**.

19.4.4 Если компонента **term** компоненты **particle** является компонентой **wildcard**, то компонента типа последовательности должна быть элементом "NamedType". Элемент "identifier" в этом элементе "NamedType" будет компонентой **elem**, а элемент "Type" в этом элементе "NamedType" должен генерироваться в результате применения подраздела 19.6 к компоненте **term** в компоненте **particle**.

19.4.5 Если компонента **term** компоненты **particle** является компонентой **model group**, то компонента типа последовательности должна быть элементом "Type" и она должна генерироваться путем применения подраздела 19.6 к компоненте **term** в компоненте **particle**.

19.4.6 Конечная команда кодирования **UNTAGGED** должна присваиваться типу последовательности.

19.5 Строка символов, используемая при генерировании элемента "identifier" в элементе "NamedType" и соответствующая компоненте **particle**, должна быть:

- строкой символов "содержание", если компонента **particle** является компонентой **content type** в компоненте **complex type definition**;
- компонентой **name** в компоненте **element declaration**, если компонента **term** компоненты **particle** является компонентой **element declaration**;
- компонентой **name** в компоненте **model group definition**, если компонента **term** компоненты **particle** является компонентой **model group** в компоненте **model group definition**;
- строкой символов **"sequence"**, если компонента **term** в компоненте **particle** является компонентой **model group** с компонентой **compositor** в компоненте **sequence**, не связанной с компонентой **model group definition**;

- e) строкой символов **"choice"**, если компонента **term** в компоненте **particle** является компонентой **model group** с компонентой **compositor** в компоненте **choice**, не связанной с компонентой **model group definition**;
- f) строкой символов **"elem"**, если компонента **term** в компоненте **particle** является компонентой **wildcard** (метасимвол).

19.6 Элемент **"Type"** в элементе **"NamedType"**, соответствующий компоненте **particle** (см. подраздел 19.3), или элемент **"Type"** в элементе **"NamedType"** в элементе **"SequenceOfType"**, соответствующий компоненте **particle** (см. подраздел 19.4), должен быть:

- a) определением типа ASN.1 (элемент **"DefinedType"**), генерируемым в результате применения подраздела 10.2 к присвоению типа ASN.1, генерируемому путем применения раздела 14 к компоненте **element declaration**, если компонента **term** в компоненте **particle** является компонентой **element declaration** верхнего уровня, не представляющей собой головной элемент группы подстановки элементов;
- b) определением типа ASN.1 (элементом **"DefinedType"**), генерируемым путем применения подраздела 10.2 к присвоению типа ASN.1, генерируемому в результате применения раздела 31 к компоненте **element declaration**, если компонента **term** в компоненте **particle** является компонентой **element declaration** верхнего уровня, представляющей собой головной элементы группы подстановки элементов;
- c) определением типа ASN.1, генерируемым путем применения раздела 14 к компоненте **element declaration**, если компонента **term** в компоненте **particle** является локальной компонентой **element declaration**;
- d) определением типа ASN.1 (элемент **"DefinedType"**), генерируемым путем применения подраздела 10.2 к присвоению типа ASN.1, генерируемому в результате применения раздела 17 к компоненте **model group definition**, если компонента **term** в компоненте **particle** является компонентой **model group** в компоненте **model group definition**;
- e) определением типа ASN.1, генерируемым путем применения раздела 18 к компоненте **model group**, если компонента **term** в компоненте **particle** является компонентой **model group**, не связанной с компонентой **model group definition**;
- f) определением типа ASN.1, генерируемым путем применения раздела 21 к компоненте **wildcard**, если компонента **term** в компоненте **particle** является компонентой **wildcard**.

20 Отображение компонент **complex type definition** (определение сложного типа)

20.1 Данный раздел применяется при инициации явным образом другими разделами настоящей Рекомендации | Международного стандарта для генерирования присвоения типа ASN.1 или определения типа ASN.1, соответствующих компоненте **complex type definition** (определение сложного типа).

20.2 Компонента **complex type definition** верхнего уровня должна отображаться в присвоении типа ASN.1. Элемент **"typereference"** в элементе **"TypeAssignment"** должен генерироваться путем применения подраздела 10.3 к компоненте **name** в компоненте **complex type definition**, а элемент **"Type"** в элементе **"TypeAssignment"** должен быть определением типа ASN.1, как указано в подразделах 20.4–20.11.

20.3 Анонимная компонента **complex type definition** должна отображаться в определении типа ASN.1, как указано в подразделах 20.4–20.11.

20.4 Определение типа ASN.1 должно быть последовательным типом ASN.1. Нулевое или большее число компонент должно добавляться к последовательному типу ASN.1 в заданном порядке, как описано в следующих подразделах.

20.5 Если компонента **content type** компоненты **complex type definition** является смешанной (**mixed**) содержательной моделью, то к последовательному типу ASN.1 должна добавляться компонента. Элемент **"identifier"** в элементе **"NamedType"** этой компоненты должен быть элементом ASN.1 **embed-values**, а элемент **"Type"** в элементе **"NamedType"** должен быть типом последовательности, компонентой которого должна быть **"Type"**, генерируемая путем применения раздела 23 к встроенному типу данных **XSD xsd:string**. Конечная команда кодирования **EMBED-VALUES** должна присваиваться последовательному типу ASN.1.

20.6 Если компонента **content type** в компоненте **complex type definition** является компонентой **particle**, компонента которой **term** является компонентой **model group** с компонентой **compositor** в компоненте **all**, к последовательному типу ASN.1 должна добавляться та или иная компонента. Элемент **"identifier"** в элементе **"NamedType"** этой компоненты должен быть элементом ASN.1 **order**, а элемент **"Type"** в элементе **"NamedType"** должен быть типом последовательности, компонентой которого должен быть элемент **"EnumeratedType"** (перечислимый тип). Для каждой компоненты **particle** в компоненте **model group** (компонента которой **term** всегда является компонентой **element declaration**) элемент **"EnumerationItem"** (элемент перечисления), который является элементом **"identifier"**, идентичным элементу **"identifier"** в элементе **"NamedType"** и соответствующим каждой компоненте **particle**, должен по порядку прибавляться к элементу **"Enumeration"**. Конечная команда кодирования **USE-ORDER** должна присваиваться последовательному типу ASN.1.

ПРИМЕЧАНИЕ. – Элементы **"identifier"** в элементах **"NamedType"**, отображаемые из компонент **particle**, генерируются (путем применения подраздела 10.3), по мере того как каждая компонента прибавляется к последовательному типу. Поэтому, если даже компонента **order** помещается на место, которое по тексту предшествует местам этих компонент в последовательности типа ASN.1, генерирование компоненты **order** может быть завершено только после того, как все компоненты **particle** будут отражены на компоненты последовательности.

20.7 Если у компоненты **complex type definition** есть компоненты **attribute use** (использование атрибута), то компоненты, генерируемые путем применения раздела 22 к компонентам **attribute use**, должны добавляться к последовательному типу ASN.1 по порядку на основе компоненты **target namespace** (целевое пространство имен) и компоненты **name** в компоненте **attribute declaration** каждой компоненты **attribute use**. Компоненты **attribute use** сначала должны быть упорядочены с помощью компоненты **target namespace** в компоненте **attribute declaration** (с ключевым словом **absent** (отсутствует), предшествующим всем именам пространства имен, отсортированным в возрастающем лексикографическом порядке), а затем с помощью компоненты **name** в компоненте **attribute declaration** в рамках каждой компоненты **target namespace** (также в возрастающем лексикографическом порядке).

20.8 Если компонента **complex type definition** имеет компоненту **attribute wildcard** (метасимвол атрибута), то компонента, генерируемая из компоненты **attribute wildcard** (см. подраздел 21.3), должна прибавляться к последовательному типу ASN.1.

20.9 Если компонента **content type** компоненты **complex type definition** является компонентой **particle**, то применяется один из следующих четырех подразделов.

20.9.1 Если компонента **term** компоненты **particle** является компонентой **model group** с компонентой **compositor** компоненты **sequence**, чьи компоненты **min occurs** и **max occurs** обе являются единицами, то по порядку для каждой компоненты **particle** в компоненте **model group** компонента, генерируемая путем применения раздела 19 к компоненте **particle** в компоненте **model group**, должна добавляться к последовательному типу ASN.1.

20.9.2 Если компонента **term** компоненты **particle** является компонентой **model group** с компонентой **compositor** компоненты **sequence**, чьи компоненты **min occurs** и **max occurs** обе не являются единицами, то компонента, генерируемая путем применения раздела 19 к компоненте **particle** в компоненте **content type**, должна добавляться к последовательному типу ASN.1.

20.9.3 Если компонента **term** компоненты **particle** является компонентой **model group** с компонентой **compositor** компоненты **all**, то для каждой компоненты **particle** в компоненте **model group** по порядку компонента, генерируемая путем применения раздела 19 к компоненте **particle** в компоненте **model group**, должна добавляться к последовательному типу ASN.1. Если компонента **particle** в компоненте **content type** компоненты **complex type definition** имеет компоненту **min occurs**, являющуюся нулем, то каждая из компонент **particle** компоненты **model group** с компонентой **min occurs**, равной единице, должна отображаться так, как если бы она имела нулевую компоненту **min occurs**.

20.9.4 Если компонента **term** компоненты **particle** является компонентой **model group** с компонентой **compositor** компоненты **choice**, то компонента, генерируемая путем применения раздела 19 к компоненте **particle** в компоненте **content type**, должна добавляться к последовательному типу ASN.1.

20.10 Если компонента **content type** компоненты **complex type definition** является компонентой **simple type definition**, то к последовательному типу ASN.1 должна добавляться некая компонента. Элемент "identifier" в элементе "NamedType" этой компоненты должен генерироваться путем применения подраздела 10.3 к строке символов "base", а элемент "Type" в элементе "NamedType" должен быть определением типа ASN.1, генерируемым путем применения раздела 23 к компоненте **content type**. Конечная команда кодирования **UNTAGGED** должна присваиваться этой компоненте.

20.11 Если компонента **content type** компоненты **complex type definition** обладает свойством **empty**, то к последовательному типу ASN.1 не должны добавляться дополнительные компоненты.

21 Отображение компонент wildcard (метасимвол)

21.1 Данный раздел применяется при инициации явным образом другими разделами настоящей Рекомендации | Международного стандарта для генерирования присвоения типа ASN.1 или определения типа ASN.1, соответствующих атрибуту **simple type definition**.

21.2 Компонента **wildcard**, которая является компонентой **term** компоненты **particle**, должна отображаться в определении типа ASN.1, генерируемом путем применения раздела 23 к встроенному типу данных XSD **xsd:string**. Конечная команда кодирования **ANY-ELEMENT** должна присваиваться определению типа ASN.1.

21.3 Компонента **wildcard**, являющаяся компонентой **attribute wildcard** компоненты **complex type**, должна отображаться в элементе "NamedType". Элемент "identifier" в элементе "NamedType" должен генерироваться путем применения подраздела 10.3 к строке символов "attr", а элемент "Type" в элементе "NamedType" должен быть типом последовательности. Компонента типа последовательности должна быть элементом "Type", генерируемым путем применения раздела 23 к встроенному типу данных XSD **xsd:string**. К типу последовательности должно применяться следующее определяемое пользователем ограничение:

(CONSTRAINED BY

{/* Каждый элемент должен отвечать формату "AnyAttributeFormat", описанному в Рекомендации МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */})

Конечная команда кодирования **ANY-ATTRIBUTES** должна присваиваться типу последовательности.

21.4 Если компонента **wildcard** имеет компоненту **namespace constraint**, то она должна отображаться в элементе "NameSpaceRestriction" (ограничение пространства имен) в команде кодирования **ANY-ELEMENT** или **ANY-ATTRIBUTES**.

22 Отображение компонент attribute use (использование атрибута)

22.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования элемента "NamedType" ASN.1, соответствующего компоненте **attribute use** (использование атрибута).

22.2 Компонента **attribute use** должна отображаться в элементе "NamedType".

22.3 Элемент "identifier" в элементе "NamedType" должен генерироваться путем применения подраздела 10.3 к компоненте **name** в компоненте **attribute declaration** компоненты **attribute use**, а элементом "Type" в элементе "NamedType" должно быть:

- определение типа ASN.1 (элемент "DefinedType"), генерируемое путем применения подраздела 10.2 к присвоению типа ASN.1, генерируемому путем применения раздела 15 к компоненте **attribute use**, если компонента **attribute declaration** имеет компоненту **attribute declaration** верхнего уровня;
- определение типа ASN.1, генерируемое в результате применения раздела 15 к компоненте **attribute declaration**, если компонента **attribute use** имеет локальную компоненту **attribute declaration**.

22.4 Если либо компонента **attribute use**, либо ее компонента **attribute declaration** имеет атрибут **value constraint**, то элемент "NamedType" должен сопровождаться ключевым словом **DEFAULT** (ПО УМОЛЧАНИЮ) и элементом "Value" (значение), генерируемым путем применения раздела 16 либо к значению компоненты **value constraint** компоненты **attribute use** (если компонента **attribute use** имеет атрибут **value constraint**), либо к значению компоненты **value constraint** ее компоненты **attribute declaration** (в противном случае).

22.5 Если либо компонента **attribute use**, либо ее компонента **attribute declaration** имеет компоненту **value constraint**, являющуюся значением со свойством **fixed**, то ограничение единичного значения ASN.1 с элементом "Value", идентичным элементу "Value", следующему за ключевым словом **DEFAULT**, должно добавляться к элементу "NamedType".

22.6 Если компонента **attribute use** не имеет свойства **required** и ни компонента **attribute use**, ни ее компонента **attribute declaration** не имеют компоненты **value constraint**, то элемент "NamedType" должен сопровождаться ключевым словом **OPTIONAL** (ФАКУЛЬТАТИВНЫЙ).

22.7 Конечная команда кодирования **ATTRIBUTE** должна присваиваться элементу "Type" в элементе "NamedType".

23 Отображение использований компонент simple type definition (определение простого типа) и complex type definition (определение сложного типа)

23.1 Этот раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего использованию компонент **simple type definition** или **complex type definition**. Рассматривается их использование как компонент **type definition** для компонент **element declaration**, которые не имеют заменяемой компоненты **type definition** (см. подраздел 14.6), не обладают свойством **nillable** и могут иметь или не иметь компоненту **value constraint**.

23.2 Использование компоненты **simple type definition** верхнего уровня, являющейся встроенным типом данных XSD, должно отображаться так, как описано в разделе 11.

23.3 Использование определяемой пользователем компоненты **simple type definition** верхнего уровня должно отображаться в определении типа ASN.1 (элемент "DefinedType"), генерируемом путем применения подраздела 10.2 к присвоению типа ASN.1, генерируемому путем применения раздела 13 к компоненте **simple type definition**.

23.4 Использование компоненты **complex type definition** верхнего уровня должно отображаться в определении типа ASN.1 (элемент "DefinedType"), генерируемом путем применения подраздела 10.2 к присвоению типа ASN.1, генерируемому путем применения раздела 20 к компоненте **complex type definition**.

23.5 Использование анонимной компоненты **simple type definition** не отличается от использования самой компоненты **simple type definition** и должно отображаться так, как указано в разделе 13 для компоненты **simple type definition**.

23.6 Использование анонимной компоненты **complex type definition** не отличается от использования самой компоненты **complex type definition** и должно отображаться так, как указано в разделе 20 для компоненты **complex type definition**.

23.7 Если компонента **simple type definition** или **complex type definition** используется как компонента **type definition** компоненты **element declaration** с компонентой **value constraint**, то конечная команда кодирования **DEFAULT-FOR-EMPTY** должна присваиваться определению типа ASN.1, и применяется один из трех следующих подразделов.

23.7.1 Для компоненты **simple type definition** элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению в компоненте **value constraint**, рассматриваемому как значение в пространстве значений компоненты **simple type definition**.

23.7.2 Для компоненты **complex type definition**, чья компонента **content type** является компонентой **simple type definition**, элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемому как значение в пространстве значений компоненты **simple type definition**.

23.7.3 Для компоненты **complex type definition** с содержательным типом со свойством **mixed** элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению в компоненте **value constraint**, рассматриваемому как значение в пространстве значений типа **xsd:string** с компонентой **whiteSpace preserve**.

23.8 Если компонента **simple type definition** или **complex type definition** используется как компонента **type definition** компоненты **element declaration** с компонентой **value constraint**, которая является значением со свойством **fixed**, то применяется один из следующих подразделов.

23.8.1 Для компоненты **simple type definition** к определению ASN.1 должно добавляться ограничение единичного значения ASN.1 с элементом "Value", идентичным элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**.

23.8.2 Для компоненты **complex type definition**, чья компонента **content type** является компонентой **simple type definition**, внутреннее ограничение подтипа ASN.1 должно добавляться к определению ASN.1, а к компоненте со свойством **base** будет применяться ограничение единичного значения с элементом "Value", идентичным элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**.

23.8.3 Для компоненты **complex type definition** с содержательным типом со свойством **mixed** внутреннее ограничение подтипа ASN.1 должно добавляться к определению ASN.1 и должно применяться:

- a) к компоненте **embed-values**, ограничению единичного значения ASN.1 с элементом "Value", заключающимся в единичном появлении элемента "Value", идентичного элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**;
- b) к каждой компоненте, которая является **OPTIONAL** и не имеет конечной команды кодирования **ATTRIBUTE**, ключевого слова **ABSENT**; и
- c) к каждой компоненте, тип которой – это тип последовательности; ограничение **SIZE(0)**.

24 Отображение особых использований компонент simple type definition и complex type definition (заменяемых)

24.1 Этот раздел применяется при инициации явным образом другими разделами настоящей Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего компоненте **simple type definition** или **complex type definition**, используемой как компонента **type definition** компонент **element declaration**, которые имеют заменяемую компоненту **type definition** (см. подраздел 14.6), не имеют свойства **nillable** и могут иметь или не иметь компоненту **value constraint**.

24.2 Использование компоненты **simple type definition** (например, STD) или **complex type definition** (например, CTD) будет отображаться в типе выбора ASN.1.

24.3 Одна альтернатива должна добавляться к типу выбора ASN.1 для STD или CTD как таковой, и одна альтернатива должна добавляться для каждой определяемой пользователем компоненты **simple type definition** или **complex type definition** верхнего уровня в схеме XSD исходного текста, которая получена путем ограничения или расширения (прямо или косвенно) из STD или CTD.

24.4 Для каждой альтернативы элемент "identifier" в элементе "NamedType" должен генерироваться путем применения подраздела 10.3 к компоненте **name** для компоненты **simple type definition** или **complex type definition**, соответствующих альтернативе, а элемент "Type" в элементе "NamedType" должен быть определением типа ASN.1, генерируемым путем применения раздела 23 к компоненте **simple type definition** или **complex type definition**.

24.5 Первая альтернатива, добавляемая к типу выбора, должна соответствовать STD или CTD как таковой. Последующие альтернативы должны добавляться к типу выбора в порядке, определяемом компонентами **target namespace** и **name** компонент **simple type definition** и **complex type definition**. Определения типов должны быть сначала упорядочены с помощью компоненты **target namespace** (с пространством имен со свойством **absent**, предшествующим всем именам пространства имен, отсортированным в возрастающем лексикографическом порядке), а затем с помощью компоненты **name** (также в возрастающем лексикографическом порядке) в пределах каждой компоненты **target namespace**.

24.6 Конечная команда кодирования **USE-TYPE** должна присваиваться типу выбора ASN.1.

24.7 Если имеется компонента **value constraint**, то конечная команда кодирования **DEFAULT-FOR-EMPTY** должна присваиваться каждой альтернативе в типе выбора ASN.1. Применяется один из трех следующих подразделов.

24.7.1 Если альтернатива соответствует компоненте **simple type definition**, то элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемому как значение в пространстве значений компоненты **simple type definition**.

24.7.2 Если альтернатива соответствует компоненте **complex type definition**, чья компонента **content type** является компонентой **simple type definition**, то элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемому как значение в пространстве значений компоненты **simple type definition**.

24.7.3 Если альтернатива соответствует компоненте **complex type definition** с содержательным типом со свойством **mixed**, то элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемому как значение в пространстве значений типа **xsd:string** с компонентой **whiteSpace preserve**.

24.8 Если имеется компонента **value constraint**, являющаяся значением со свойством **fixed**, то внутреннее ограничение подтипа должно добавляться к типу выбора ASN.1. Применяется один из следующих трех подразделов.

24.8.1 Если альтернатива соответствует компоненте **simple type definition**, то внутреннее ограничение подтипа следует применять к альтернативе ограничение единичного значения ASN.1 с элементом "Value", идентичным элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**.

24.8.2 Если альтернатива соответствует компоненте **complex type definition**, чья компонента **content type** является компонентой **simple type definition**, то внутреннее ограничение подтипа следует применять к альтернативе другое внутреннее ограничение подтипа ASN.1, которое применяет к компоненте со свойством **base** ограничение единичного значения с элементом "Value", идентичным элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**.

24.8.3 Если альтернатива соответствует компоненте **complex type definition** с содержательным типом со свойством **mixed**, то внутреннее ограничение подтипа будет применять к альтернативе другое внутреннее ограничение подтипа ASN.1, которое применяет:

- a) к компоненте со свойством **embed-values** ограничение единичного значения ASN.1 с элементом "Value", заключающееся в единичном появлении элемента "Value", идентичного элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**;
- b) к каждой компоненте, имеющей свойство **OPTIONAL** (**ФАКУЛЬТАТИВНАЯ**) и не имеющей конечной команды кодирования **ATTRIBUTE**, ключевое слово **ABSENT**; и
- c) к каждой компоненте, чей тип является типом последовательности, ограничение **SIZE (0)**.

25 Отображение особых использований компонент **simple и **complex type definitions** (**substitutable, nillable**)**

25.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего компоненте **simple type definition** или **complex type definition**, используемой как компонента **type definition** для компонент **element declaration**, которые имеют заменяемую компоненту **type definition** (см. подраздел 14.6), обладают свойством **nillable** (обнуляемые) и могут иметь или не иметь компоненту **value constraint**.

25.2 Использование компоненты **simple type definition** (например, STD) или **complex type definition** (например, CTD) должно отображаться в типе выбора ASN.1.

25.3 Одна альтернатива должна добавляться к типу выбора ASN.1 для STD или CTD как таковой, и одна альтернатива должна добавляться для каждой определяемой пользователем компоненты **simple type definition** и **complex type definition** верхнего уровня в схеме XSD исходного текста, который получен путем ограничения и расширения (прямо или косвенно) из STD или CTD.

25.4 Для каждой альтернативы элемент "identifier" в элементе "NamedType" должен генерироваться путем применения подраздела 10.3 к компоненте **name** компоненты **simple type definition** или **complex type definition**, соответствующей альтернативе, а элемент "Type" в элементе "NamedType" должен быть определением типа ASN.1 (элемент "DefinedType"), генерируемым путем применения подраздела 10.2 к присвоению типа ASN.1, генерируемому в результате применения раздела 30 к компоненте **simple type definition** или **complex type definition**.

25.5 Первая альтернатива, добавляемая к типу выбора, должна соответствовать STD или CTD как таковой. Последующие альтернативы должны добавляться к типу выбора в порядке на основе компонент **target namespace** и **name** компонент **simple type definition** и **complex type definition**. Определения типов должны сначала упорядочиваться согласно компоненте **target namespace** (с пространством имен со свойством **absent**, предшествующим всем именам пространства имен, отсортированным в возрастающем лексикографическом порядке), а затем согласно компоненте **name** (также в возрастающем лексикографическом порядке) в пределах каждой компоненты **target namespace**.

25.6 Конечная команда кодирования **USE-TYPE** должна присваиваться типу выбора ASN.1.

25.7 Если имеется компонента **value constraint**, то конечная команда кодирования **DEFAULT-FOR-EMPTY** должна присваиваться каждой альтернативе типа выбора ASN.1. Применяется один из следующих трех подразделов.

25.7.1 Если альтернатива соответствует компоненте **simple type definition**, то элемент "Value" в конечной команде присвоения **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемой как значение в пространстве значений компоненты **simple type definition**.

25.7.2 Если альтернатива соответствует компоненте **complex type definition**, компонента которой **content type** является компонентой **simple type definition**, то элемент "Value" в конечной команде присвоения **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемой как значение в пространстве значений компоненты **simple type definition**.

25.7.3 Если альтернатива соответствует компоненте **complex type definition** с содержательным типом со свойством **mixed**, то элемент "Value" в конечной команде присвоения **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемому как значение в пространстве значений типа **xsd:string** с компонентой **whiteSpace preserve**.

25.8 Если имеется компонента **value constraint**, являющаяся значением со свойством **fixed**, то внутреннее ограничение подтипа ASN.1 должно добавляться к типу выбора ASN.1. Применяется один из следующих трех подразделов.

25.8.1 Если альтернатива соответствует компоненте **simple type definition**, то внутреннее ограничение подтипа должно применять к альтернативе (которая является последовательным типом ASN.1 с конечной командой кодирования **USE-NIL**) другое внутреннее ограничение подтипа ASN.1, которое в свою очередь будет применять к компоненте **content** ключевое слово **PRESENT** и ограничение единичного значения ASN.1 с элементом "Value", идентичным элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**.

25.8.2 Если альтернатива соответствует компоненте **complex type definition**, компонента которой **content type** является компонентой **simple type definition**, то внутреннее ограничение подтипа будет применять к альтернативе (которая является последовательным типом ASN.1 с конечной командой кодирования **USE-NIL**) другое внутреннее ограничение подтипа ASN.1, которое применяет к компоненте со свойством **content** ключевое слово **PRESENT** и ограничение единичного значения ASN.1 с элементом "Value", идентичным элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**.

25.8.3 Если альтернатива соответствует компоненте **complex type definition** с содержательным типом **mixed**, то внутреннее ограничение подтипа должно применять к альтернативе (которая является последовательным типом ASN.1 с конечной командой кодирования **USE-NIL**) другое внутреннее ограничение подтипа ASN.1, которое применяет:

- a) к компоненте со свойством **embed-values** – ограничение единичного значения ASN.1 с элементом "Value", заключающееся в единичном появлении элемента "Value", идентичного элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**;
- b) к компоненте со свойством **content** (которая является последовательным типом ASN.1) ключевое слово **PRESENT** и другое внутреннее ограничение подтипа, которое применяет ключевое слово **ABSENT** к каждой из его компонент, которая обладает свойством **OPTIONAL**, а ограничение **SIZE (0)** – к каждой из его компонент, типом которой является тип последовательности;
- c) ключевое слово **ABSENT** к каждой компоненте, которая имеет свойство **OPTIONAL** и не имеет конечной команды кодирования **ATTRIBUTE**; и
- d) ограничение **SIZE (0)** к каждой компоненте, тип которой – это тип последовательности.

26 Отображение особых использований компонент **simple type definitions (nillable)**

26.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего компоненте **simple type definition**, используемой как компонента **type definition** в компонентах **element declaration**, которые не имеют заменяемой компоненты **type definition** (см. подраздел 14.6), обладают свойством **nillable** (обнуляемые) и могут иметь или не иметь компоненты **value constraint**.

26.2 Использование компоненты **simple type definition** должно отображаться в последовательном типе ASN.1 с одной компонентой со свойством **OPTIONAL** (**ФАКУЛЬТАТИВНАЯ**).

26.3 Элемент "identifier" в элементе "NamedType" компоненты должен обладать свойством **content**, а элемент "Type" в элементе "NamedType" должен быть определением типа ASN.1, генерируемым путем применения раздела 23 к компоненте **simple type definition**.

26.4 Конечная команда кодирования **USE-NIL** должна присваиваться последовательному типу ASN.1.

26.5 Если имеется компонента **value constraint**, то конечная команда кодирования **DEFAULT-FOR-EMPTY** должна присваиваться последовательному типу ASN.1. Элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**.

26.6 Если имеется компонента **value constraint**, которая является значением со свойством **fixed**, то внутреннее ограничение подтипа ASN.1 должно добавляться к последовательному типу ASN.1. Внутреннее ограничение подтипа должно применять к компоненте **content** ограничение единичного значения ASN.1 с элементом "Value", идентичным элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**. Внутреннее ограничение подтипа также должно применять ключевое слово **PRESENT** к компоненте **content**.

27 Отображение особых использований компонент **complex type definitions (nillable)**

27.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего компоненте **complex type definition**, используемой как компонента **type definition** для компонент **element declaration**, которые не имеют заменяемой компоненты **type definition** (см. подраздел 14.6), обладают свойством **nillable** (обнуляемые) и могут иметь или не иметь компоненты **value constraint**.

27.2 Использование компоненты **complex type definition** должно отображаться в последовательном типе ASN.1. Одна или несколько компонент должны добавляться к последовательному типу в заданном порядке, как это описано в следующих подразделах.

27.3 Если компонента **content type** компоненты **complex type definition** является содержательной моделью со свойством **mixed**, то компонента со свойством **embed-values** должна добавляться к последовательному типу ASN.1, как указано в подразделе 20.5.

27.4 Если компонента **content type** компоненты **complex type definition** является компонентой **particle**, у которой компонента **term** является компонентой **model group** с компонентой **compositor** со свойством **all**, то компонента **order** должна добавляться к последовательному типу ASN.1, как указано в подразделе 20.6.

27.5 Если компонента **complex type definition** имеет компоненты **attribute use**, то компоненты, отображаемые из компонент **attribute use**, должны добавляться к последовательному типу ASN.1, как указано в подразделе 20.7.

27.6 Если компонента **complex type definition** имеет компоненту **attribute wildcard**, то компонента, генерируемая из компоненты **attribute wildcard**, должна добавляться к последовательному типу ASN.1, как описано в подразделе 20.8.

27.7 Если компонента **content type** компоненты **complex type definition** является компонентой **particle**, то применяется один из следующих двух подразделов.

27.7.1 Если компонента **term** компоненты **particle** является компонентой **model group** с компонентой **compositor** со свойством **sequence** или **choice**, то компонента **OPTIONAL** должна добавляться к последовательному типу ASN.1. Элемент "identifier" в элементе "NamedType" этой компоненты должен генерироваться путем применения подраздела 10.3 к строке символов "content", а элемент "Type" в элементе "NamedType" должен быть последовательным типом ASN.1 с единичной компонентой, который должен генерироваться путем применения раздела 19 к компоненте **particle** в компоненте **content type**.

27.7.2 Если компонента **term** компоненты **particle** является компонентой **model group** с компонентой **compositor** со свойством **all**, то компонента **OPTIONAL** должна добавляться к последовательному типу ASN.1. Элемент "identifier" в элементе "NamedType" компоненты должен генерироваться путем применения подраздела 10.3 к строке символов "content", а элемент "Type" в элементе "NamedType" должен быть последовательным типом ASN.1. Для каждой компоненты **particle** в компоненте **model group** компонента, генерируемая путем применения раздела 19 к компоненте **particle** в компоненте **model group**, должна добавляться к внутреннему последовательному типу ASN.1. Если компонента **particle** в компоненте **content type** компоненты **complex type definition** имеет нулевую компоненту **min occurs**, то каждая из компонент **particle** компоненты **model group** с компонентой **min occurs**, равной единице, должна отображаться так, как если бы она имела нулевую компоненту **min occurs**.

27.8 Если компонента **content type** компоненты **complex type definition** является компонентой **simple type definition**, то компонента **OPTIONAL** должна добавляться к последовательному типу ASN.1. Элемент "identifier" в элементе "NamedType" компоненты должен генерироваться путем применения подраздела 10.3 к строке символов "content", а элемент "Type" в элементе "NamedType" должен быть определением типа ASN.1, генерируемым путем применения раздела 23 к компоненте **content type**.

27.9 Если компонента **content type** компоненты **complex type definition** обладает свойством **empty** (пустая), то к последовательному типу ASN.1 дополнительные компоненты добавлять не следует.

27.10 Конечная команда кодирования **USE-NIL** должна присваиваться последовательному типу ASN.1.

27.11 Если имеется компонента **value constraint**, то конечная команда кодирования **DEFAULT-FOR-EMPTY** должна присваиваться последовательному типу ASN.1. Применяется один из двух следующих подразделов.

27.11.1 Если компонента **content type** в компоненте **complex type definition** является компонентой **simple type definition**, то элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемому как значение в пространстве значений компоненты **simple type definition**.

27.11.2 Если компонента **content type** в компоненте **complex type definition** является содержательным типом **mixed**, то элемент "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY** должен генерироваться путем применения раздела 16 к значению компоненты **value constraint**, рассматриваемому как значение в пространстве значений типа **xsd:string** с компонентой **whiteSpace preserve**.

27.12 Если имеется компонента **value constraint**, являющаяся значением со свойством **fixed**, то внутреннее ограничение подтипа ASN.1 должно добавляться к последовательному типу ASN.1. Это внутреннее ограничение подтипа будет применять ключевое слово **PRESENT** к компоненте со свойством **content**. Применяется один из двух следующих подразделов.

27.12.1 Если компонента **content type** в компоненте **complex type definition** является компонентой **simple type definition**, то внутреннее ограничение подтипа должно применяться к компоненте со свойством **content** ограничение единичного значения ASN.1 с элементом "Value", идентичным элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**.

27.12.2 Если компонента **content type** в компоненте **complex type definition** является содержательным типом со свойством **mixed**, то внутреннее ограничение подтипа должно применяться:

- а) к компоненте со свойством **embed-values** – ограничение единичного значения ASN.1 с элементом "Value", заключающееся в единичном появлении элемента "Value", идентичного элементу "Value" в конечной команде кодирования **DEFAULT-FOR-EMPTY**;
- б) ключевое слово **ABSENT** к каждой компоненте в компоненте со свойством **content** (последовательный тип ASN.1), которая является **OPTIONAL**;
- в) ограничение **SIZE(0)** к каждой компоненте в компоненте со свойством **content** (последовательный тип ASN.1), типом которой является тип последовательности.

28 Отображение особых использований компонент element declarations (головной элемент группы замены элементов)

28.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего компоненте **element declaration** верхнего уровня, являющейся головным элементом группы замены элементов и используемой как компонента **term** компонент **particles**.

28.2 Использование компоненты **element declaration** верхнего уровня должно отображаться в типе выбора ASN.1.

28.3 Одна альтернатива должна добавляться к типу выбора ASN.1 для компоненты **element declaration** верхнего уровня (например, H) как таковой, и одна альтернатива должна добавляться для каждой компоненты **element declaration** верхнего уровня в схеме XSD исходного текста, которая не обладает свойством **abstract** и компонента которой **substitution group affiliation** (присоединение к группе замены) является компонентой H.

28.4 Для каждой альтернативы элемент "identifier" в элементе "NamedType" должен генерироваться путем применения подраздела 10.3 к компоненте **name** в компоненте **element declaration** верхнего уровня, соответствующей данной альтернативе, а элемент "Type" в элементе "NamedType" должен быть определением типа ASN.1 (элемент "DefinedType"), генерируемым путем применения подраздела 10.2 к присвоению типа ASN.1, генерируемому путем применения раздела 14 к компоненте **element declaration** верхнего уровня.

ПРИМЕЧАНИЕ. – В XSD членство в группе замены носит транзитивный характер, то есть члены группы замены ESG1, главный элемент которой является членом другой группы замены ESG2, все являются также членами группы ESG2.

28.5 Альтернативы следует добавлять к типу выбора в порядке, который основан на компоненте **target namespace** и компоненте **name** компонент **element declaration** верхнего уровня. Компоненты **element declaration** сначала следует упорядочить согласно компоненте **target namespace** (с пространством имен со свойством **absent**, предшествующим всем именам пространства имен в возрастающем лексикографическом порядке), а затем согласно компоненте **name** (также в возрастающем лексикографическом порядке) в рамках каждой компоненты **target namespace**.

ПРИМЕЧАНИЕ. – Компонента **element declaration**, являющаяся головным элементом группы замены элементов, упорядочивается вместе с другими компонентами **element declaration**, которые принадлежат к группе замены элементов.

28.6 Конечная команда кодирования **UNTAGGED** должна присваиваться типу выбора.

29 Генерирование особых присвоений типов ASN.1 для компонент **element declarations**

29.1 Этот раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего определяемым пользователем компонентам **simple type definition** или **complex type definition** верхнего уровня, которые имеют заменяемую компоненту **type definition** (см. подраздел 14.6) или обладают свойством **nillable**.

29.2 Данный раздел иницируется другими разделами для заданного сочетания элементов, а именно:

- компонент **simple type definition** или **complex type definition**;
- какие из компонент **element declaration** имеют заменяемую компоненту **type definition** (см. подраздел 14.6);
- обладает ли компонента **element declaration** свойством **nillable**; и
- имеет ли компонента **element declaration** компоненту **value constraint** и вид и значение компоненты **value constraint**;

и для генерирования присвоения типа ASN.1 (называемого "особым присвоением типа ASN.1 (для описаний элементов)") для сочетания упомянутых выше элементов.

29.3 Одно и только одно особое присвоение типа ASN.1 будет генерироваться для каждой отдельной комбинации упомянутых выше элементов, которая фактически встречается в одном или нескольких применениях данного раздела при отображении схемы XSD исходного текста.

ПРИМЕЧАНИЕ. – Например, если две или более компонент **element declarations** в большой схеме XSD имеют идентичные компоненты **type definitions**, обе обладают свойством **nillable** и обе имеют компоненту **value constraint**, являющуюся значением со свойством **default** и одним и тем же значением, то генерируется одно особое присвоение типа ASN.1. Имя ссылки на тип этого присвоения типа будет встречаться в элементе "Type" в элементах "TypeAssignment", соответствующих обоим компонентам **element declarations**.

29.4 Термин "ассоциированное присвоение типа ASN.1" обозначает присвоение типа ASN.1, отображаемое от компоненты **simple type definition** или **complex type definition**, являющейся компонентой **type definition** компоненты **element declaration**, для которой генерируется особое присвоение типа ASN.1 путем применения, соответственно, раздела 13 или раздела 20.

ПРИМЕЧАНИЕ. – Всякое особое присвоение типа ASN.1 имеет ассоциированное присвоение типа ASN.1, поскольку этот раздел применяется, только когда компонента **type definition** компоненты **element declaration** является определяемой пользователем компонентой **simple type definition** или **complex type definition** верхнего уровня. Все такие компоненты **simple type definitions** или **complex type definition** отображаются в присвоении типов ASN.1.

29.5 Для той или иной данной компоненты **element declaration** элемент "typereference" в элементе "TypeAssignment" для особого присвоения типа ASN.1 должен создаваться путем добавления суффикса к имени ссылки на тип ассоциированного присвоения типа ASN.1 и применения подраздела 10.3 к результирующей строке символов, а элемент "Type" в элементе "TypeAssignment" должен быть определением типа ASN.1, генерируемым путем применения раздела 24 либо 25 к компоненте **simple type definition** или **complex type definition**, являющейся компонентой **type definition** компоненты **element declaration**. Применяется одно из следующих положений:

- если компонента **element declaration** не обладает свойством **nillable**, имеет заменяемую компоненту **type definition** и не имеет компоненты **value constraint**, то суффиксом должен быть "-derivations" и должен применяться раздел 24;
- если компонента **element declaration** обладает свойством **nillable**, имеет заменяемую компоненту **type definition**, не имеет компоненты **value constraint**, то суффиксом должен быть "-deriv-nillable" и должен применяться раздел 25;
- если компонента **element declaration** не обладает свойством **nillable**, имеет заменяемую компоненту **type definition** и компоненту **value constraint**, являющуюся значением со свойством **default**, то суффиксом должен быть "-deriv-default-", который сопровождается каноническим лексическим представлением (см. часть 2, 2.3.1 схемы XML консорциума W3C) значения компоненты **value constraint**, и должен применяться раздел 24;
- если компонента **element declaration** не обладает свойством **nillable**, имеет заменяемую компоненту **type definition** и компоненту **value constraint**, являющуюся значением со свойством **fixed**, то суффиксом должен быть "-deriv-fixed-", который сопровождается каноническим лексическим представлением (см. часть 2, 2.3.1 Схемы XML консорциума W3C) значения компоненты **value constraint**, и должен применяться раздел 24;

- e) если компонента **element declaration** обладает свойством **nillable**, имеет заменяемую компоненту **type definition** и компоненту **value constraint**, являющуюся значением со свойством **default**, то суффиксом должен быть **"-deriv-nillable-default-"**, за которым следует каноническое лексическое представление (см. часть 2, 2.3.1 Схемы XML консорциума W3C) значения компоненты **value constraint**, и должен применяться раздел 25;
- f) если компонента **element declaration** обладает свойством **nillable**, имеет заменяемую компоненту **type definition** и компоненту **value constraint**, являющуюся значением со свойством **fixed**, то суффиксом должен быть **"-deriv-nillable-fixed-"**, за которым следует каноническое лексическое представление (см. часть 2, 2.3.1 Схемы XML консорциума W3C) значения компоненты **value constraint**, и должен применяться раздел 25.

30 Генерирование особых присвоений типов ASN.1 для компонент **type definitions**

30.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего определяемой пользователем компоненте **simple type definition** или **complex type definition** верхнего уровня, которая принадлежит к иерархии получения компоненты **type definition** компонент **element declaration**, имеющих заменяемую компоненту **type definition** (см. подраздел 14.6) и обладающих свойством **nillable**.

30.2 Данный раздел иницируется другими разделами для той или иной конкретной компоненты **simple type definition** или **complex type definition** для генерирования присвоения типа ASN.1 (называемого "особым присвоением типа ASN.1 (для определения типа)").

30.3 Одно и только одно особое присвоение типа ASN.1 должно генерироваться для каждой компоненты **simple type definition** или **complex type definition**, которая фактически встречается в одном или нескольких применениях данного раздела при отображении схемы XSD исходного текста.

30.4 Термин "ассоциированное присвоение типа ASN.1" обозначает присвоение типа ASN.1, отображаемое от компоненты **simple type definition** или **complex type definition** путем применения, соответственно, раздела 13 или раздела 20.

30.5 Элемент "typereference" в элементе "TypeAssignment" для особого присвоения типа ASN.1 должен образовываться путем добавления суффикса **"-nillable"** к имени ссылки на тип ассоциированного присвоения типа ASN.1 и применения подраздела 10.3 к результирующей строке символов, а элемент "Type" в элементе "TypeAssignment" будет определением типа ASN.1, генерируемым путем применения либо раздела 26, либо раздела 27 соответственно к компоненте **simple type definition** или **complex type definition**.

31 Генерирование особых присвоений типов ASN.1 для групп замены элементов

31.1 Данный раздел применяется при инициации явным образом другими разделами данной Рекомендации | Международного стандарта для генерирования определения типа ASN.1, соответствующего компоненте **particle**, компонента которой **term** является компонентой **element declaration** верхнего уровня, которая представляет собой головной элемент группы замены элементов.

31.2 Данный раздел иницируется другими разделами для компоненты **element declaration** верхнего уровня, являющейся головным элементом группы замены элементов, и для генерирования присвоения типа ASN.1 (называемого "особым присвоением типа ASN.1 (для группы замены элементов)").

31.3 Одно и только одно особое присвоение типа ASN.1 должно генерироваться для каждой компоненты **element declaration** верхнего уровня, которая фактически возникает при одном или нескольких применениях данного раздела при отображении схемы XSD исходного текста.

31.4 Термин "ассоциированное присвоение типа ASN.1" обозначает присвоение типа ASN.1, отображаемое от компоненты **element declaration** верхнего уровня путем применения раздела 14.

31.5 Элемент "typereference" в элементе "TypeAssignment" для особого присвоения типа ASN.1 должен образовываться путем добавления суффикса **"-group"** к имени ссылки на тип ассоциированного присвоения типа ASN.1 и применения подраздела 10.3 к результирующей строке символов, а элемент "Type" в элементе "TypeAssignment" должен быть определением типа ASN.1, генерируемым путем применения раздела 28 к компоненте **element declaration** верхнего уровня.

Приложение А

Определение типов ASN.1, соответствующих встроенным типам данных XSD

(Данное Приложение образует неотъемлемую часть настоящей Рекомендации | Международного стандарта)

A.1 В данном Приложении дано описание модуля, определяющего типы ASN.1, которые соответствуют встроенным типам данных XSD и используются для отображения схемы языка XML консорциума W3C на систему обозначений ASN.1.

A.2 Схемой языка XML консорциума W3C определяются многие встроенные типы данных даты и времени для представления видов продолжительности, моментов времени или повторяющихся моментов времени. Хотя все они и получены из стандарта ИСО 8601, имеется ряд расширений и ограничений. Встроенные типы данных даты и времени отображаются в элементе **visibleString** (видимая строка) с определяемым пользователем ограничением со ссылкой на применяемый раздел XSD. Для обеспечения более эффективного кодирования по правилам пакетного кодирования (PER) добавляется допустимое алфавитное ограничение, поскольку определяемые пользователем ограничения не видимы при кодировании PER (и, следовательно, не используются в оптимизирующих видах кодирования).

A.3 Модулем XSD является модуль:

```
XSD {joint-iso-itu-t asn1(1) specification(0) modules(0) xsd-module(2)}
DEFINITIONS
AUTOMATIC TAGS ::=
BEGIN

/* xsd:anySimpleType */
AnySimpleType ::= XMLCompatibleString
/* xsd:anyType */
AnyType ::= SEQUENCE {
    embed-values SEQUENCE OF String,
    attr SEQUENCE
        (CONSTRAINED BY {
            /* Каждый элемент должен отвечать формату "AnyAttributeFormat", описанному
             в Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */ } ) OF String,
    elem-list SEQUENCE OF elem String
        (CONSTRAINED BY {
            /* Должен отвечать формату "AnyElementFormat", описанному
             в Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 19 */ } ) }
        (CONSTRAINED BY {
            /* Должен соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 25 */ } )
/* xsd:anyUri */
AnyURI ::= XMLStringWithNoCRLFHT
        (CONSTRAINED BY {
            /* XMLStringWithNoCRLFHT должен быть действительным URI, как определено в RFC
             IETF 2396 */ } )
/* xsd:base64Binary */
Base64Binary ::= OCTET STRING
/* xsd:булев */
Boolean ::= BOOLEAN
/* xsd:байт */
Byte ::= INTEGER (-128..127)
/* xsd:дата */
Date ::= DATE-TIME (DateOnly)
/* xsd:dateTime */
DateTime ::= DATE-TIME
/* xsd:десятичный */
Decimal ::= REAL (WITH COMPONENTS {..., base(10)})
        (ALL EXCEPT(-0 | MINUS-INFINITY | PLUS-INFINITY | NOT-A-NUMBER))
/* xsd:двойной */
```

```

Double ::= REAL (WITH COMPONENTS {
    mantissa(-9007199254740991..9007199254740991),
    base(2),
    exponent(-1075..970)})

/* xsd:продолжительность */
Duration ::= DURATION

/* xsd:ENTITIES */
ENTITIES ::= SEQUENCE (SIZE(1..MAX)) OF ENTITY

/* xsd:ENIITY */
ENTITY ::= NCName

/* xsd:float */
Float ::= REAL (WITH COMPONENTS {
    mantissa(-16777215..16777215),
    base(2),
    exponent(-149..104)})

/* xsd:gDay */
GDay ::= DATE-TIME (Day)

/* xsd:gMonth */
GMonth ::= DATE-TIME (Month)

/* xsd:gMonthDay */
GMonthDay ::= DATE-TIME (MonthDay)

/* xsd:gYear */
GYear ::= DATE-TIME (Year)

/* xsd:gYearMonth */
GYearMonth ::= DATE-TIME (YearMonth)

/* xsd:hexBinary */
HexBinary ::= OCTET STRING

/* xsd:ID */
ID ::= NCName

/* xsd:IDREF */
IDREF ::= NCName

/* xsd:IDREFS */
IDREFS ::= SEQUENCE (SIZE(1..MAX)) OF IDREF

/* xsd:int */
Int ::= INTEGER (-2147483648..2147483647)

/* xsd:целое */
Integer ::= INTEGER

/* xsd:язык */
Language ::= VisibleString (FROM ("a".."z" | "A".."Z" | "-" | "0".."9"))
    (PATTERN
        "[a-zA-Z]#{1,8}(-[a-zA-Z0-9]#{1,8})")
    /* Семантика языка описана в RFC IETF 3066 */

/* xsd:длинный */
Long ::= INTEGER (-9223372036854775808..9223372036854775807)

/* xsd:имя */

```

```

Name ::= Token (XMLStringWithNoWhitespace)
        (CONSTRAINED BY {
            /* Token должно быть Name, как определено в части 1.0, 2.3 XML
               консорциума W3C */ } )

/* xsd:NCName */
NCName ::= Name
        (CONSTRAINED BY {
            /* Name должно быть NCName, как определено в части 2 пространства имен
               XML W3C */ } )

/* xsd:negativeInteger */
NegativeInteger ::= INTEGER (MIN..-1)

/* xsd:NMTOKEN */
NMTOKEN ::= Token (XMLStringWithNoWhitespace)
        (CONSTRAINED BY {
            /* Token должно быть NMTOKEN, как определено в части 1.0, 2.3 XML
               консорциума W3C */ } )

/* xsd:NMTOKENS */
NMTOKENS ::= SEQUENCE (SIZE(1..MAX)) OF NMTOKEN

/* xsd:nonNegativeInteger */
NonNegativeInteger ::= INTEGER (0..MAX)

/* xsd:nonPositiveInteger */
NonPositiveInteger ::= INTEGER (MIN..0)

/* xsd:normalizedString */
NormalizedString ::= String (XMLStringWithNoCRLFHT)
        (CONSTRAINED BY {
            /* String должно быть normalizedString, как определено в части 2, 3.3.1
               Схемы языка XML консорциума W3C */})

/* xsd:NOTATION */
NOTATION ::= QName

/* xsd:positiveInteger */
PositiveInteger ::= INTEGER (1..MAX)

/* xsd:QName */
QName ::= SEQUENCE {
    uri   AnyURI OPTIONAL,
    name  NCName }

/* xsd:короткий */
Short ::= INTEGER (-32768..32767)

/* xsd:строка */
String ::= XMLCompatibleString

/* xsd:время */
Time ::= DATE-TIME (TimeOnly)

/* xsd:маркер */
Token ::= NormalizedString (CONSTRAINED BY {
    /* NormalizedString должно быть маркером, как определено в части 2, 3.3.2
       Схемы XML консорциума W3C */})

/* xsd:unsignedByte */
UnsignedByte ::= INTEGER (0..255)

/* xsd:unsignedInt */
UnsignedInt ::= INTEGER (0..4294967295)

/* xsd:unsignedLong */
UnsignedLong ::= INTEGER (0..18446744073709551615)

```

```

/* xsd:unsignedShort */
UnsignedShort ::= INTEGER (0..65535)

/* Определения типов ASN.1, поддерживающие отображение встроенных типов данных
Схемы XML консорциума W3C */
XMLCompatibleString ::= UTF8String (FROM(
    {0, 0, 0, 9} |
    {0, 0, 0, 10} |
    {0, 0, 0, 13} |
    {0, 0, 0, 32} .. {0, 0, 215, 255} |
    {0, 0, 224, 0} .. {0, 0, 255, 253} |
    {0, 1, 0, 0} .. {0, 16, 255, 253}))

XMLStringWithNoWhitespace ::= UTF8String (FROM(
    {0, 0, 0, 33} .. {0, 0, 215, 255} |
    {0, 0, 224, 0} .. {0, 0, 255, 253} |
    {0, 1, 0, 0} .. {0, 16, 255, 253}))

XMLStringWithNoCRLFHT ::= UTF8String (FROM(
    {0, 0, 0, 32} .. {0, 0, 215, 255} |
    {0, 0, 224, 0} .. {0, 0, 255, 253} |
    {0, 1, 0, 0} .. {0, 16, 255, 253}))

/* Определения типов данных ASN.1, поддерживающие отображение встроенных типов
данных даты и времени Схемы XML W3C */
DURATION ::= VisibleString (FROM ("0".."9" | "DHMPSTY:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.6 Схемы XML W3C */})

DATE-TIME ::= VisibleString (FROM ("0".."9" | "TZ:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.7 Схемы XML W3C */})

DateOnly ::= DATE-TIME (FROM ("0".."9" | "Z:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.9 Схемы XML W3C */})

Day ::= DATE-TIME (FROM ("0".."9" | "Z:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.13 Схемы XML W3C */})

Month ::= DATE-TIME (FROM ("0".."9" | "Z:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.14 Схемы XML W3C */})

MonthDay ::= DATE-TIME (FROM ("0".."9" | "Z:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.12 Схемы XML W3C */})

Year ::= DATE-TIME (FROM ("0".."9" | "Z:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.11 Схемы XML W3C */})

YearMonth ::= DATE-TIME (FROM ("0".."9" | "Z:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.10 Схемы XML W3C */})

TimeOnly ::= DATE-TIME (FROM ("0".."9" | "Z:+-"))
(CONSTRAINED BY {/* часть 2, 3.2.8 Схемы XML W3C */})

ENCODING-CONTROL XER
    GLOBAL-DEFAULTS MODIFIED-ENCODINGS
    GLOBAL-DEFAULTS CONTROL-NAMESPACE
        "http://www.w3.org/2001/XMLSchema-instance"
    PREFIX "xsi"
    NAMESPACE ALL, ALL IN ALL AS
        "http://www.w3.org/2001/XMLSchema"
    PREFIX "xsd"
    USE-QNAME QName
    BASE64 Base64Binary
    DECIMAL Decimal
    LIST ENTITIES, IDREFS, NMTOKENS
    EMBED-VALUES AnyType
    ANY-ATTRIBUTES AnyType.any-attributes
    ANY-ELEMENT AnyType.any-elements.*
    UNTAGGED AnyType.any-elements
    NAME AnySimpleType, AnyURI, Base64Binary, Boolean,
        Byte, Date, DateTime, Decimal, Double, Duration,

```

```
Float, GDay, GMonth, GMonthDay, GYear, GYearMonth,  
HexBinary, Int, Integer, Language, Long,  
NegativeInteger, NonNegativeInteger, NonPositiveInteger,  
NormalizedString, PositiveInteger, Short,  
String, Time, Token,  
UnsignedByte, UnsignedInt, UnsignedLong, UnsignedShort  
AS UNCAPITALIZED  
WHITESPACE AnyURI, Language, Token, DURATION, DATE-TIME COLLAPSE  
WHITESPACE NormalizedString REPLACE
```

END

Приложение В

Присвоение значений идентификаторов объектов

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Приведенное ниже значение идентификатора объекта и дескриптора объекта присваивается в данной Рекомендации | Международном стандарте:

Для модуля, определяющего типы ASN.1, соответствующие встроенным типам данных XSD:

```
{ joint-iso-itu-t asn1(1) specification(0) modules(0) xsd-module(2) }
```

"Модуль ASN.1 - XSD"

Приложение С

Примеры отображений

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В данном Приложении приводятся примеры отображений, описанных в настоящей Рекомендации | Международном стандарте, с помощью представления модуля ASN.1, соответствующего схеме XSD.

С.1 Схема, использующая компоненты simple type definitions (определение простого типа)

Приведенная ниже схема содержит примеры встроенных типов данных XSD (`xsd:string`, `xsd:decimal`, `xsd:integer`, `xsd:int`, `xsd:date`), другие компоненты simple type definition и complex type definition.

```
<?xml version="1.0" encoding="UTF-8"?>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified">
    <xsd:element name="EXAMPLES">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="personnelRecord"/>
          <xsd:element name="decimal" type="xsd:decimal"/>
          <xsd:element name="daysOfTheWeek" type="ListOfDays"/>
          <xsd:element ref="namesOfMemberNations"/>
          <xsd:element ref="fileIdentifier" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="personnelRecord">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="name" type="name"/>
          <xsd:element name="title" type="xsd:string"/>
          <xsd:element name="decimal" type="xsd:integer"/>
          <xsd:element name="dateOfHire" type="xsd:date"/>
          <xsd:element ref="nameOfSpouse"/>
          <xsd:element ref="children"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="nameOfSpouse" type="name"/>
    <xsd:complexType name="name">
      <xsd:sequence>
        <xsd:element name="givenName" type="xsd:string"/>
        <xsd:element name="initial" type="xsd:string"/>
        <xsd:element name="familyName" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="children">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="ChildInformation" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="ChildInformation">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="name" type="name"/>
          <xsd:element name="dateOfBirth" type="xsd:date"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name="ListOfDays">
      <xsd:list itemType="Day"/>
    </xsd:simpleType>
    <xsd:simpleType name="Day">
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="monday"/>
        <xsd:enumeration value="tuesday"/>
        <xsd:enumeration value="wednesday"/>
        <xsd:enumeration value="thursday"/>
        <xsd:enumeration value="friday"/>
        <xsd:enumeration value="saturday"/>
        <xsd:enumeration value="sunday"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:schema>
```

```

</xsd:simpleType>
<xsd:element name="namesOfMemberNations">
  <xsd:simpleType>
    <xsd:list itemType="xsd:string"/>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="fileIdentifier">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element name="serialNumber" type="xsd:int"/>
      <xsd:element name="relativeName" type="xsd:string"/>
      <xsd:element ref="unidentified"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name="unidentified">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="xsd:anyType"/>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

C.2 Соответствующие определения ASN.1

Ниже приводится соответствующая спецификация ASN.1 и проверка правильности тех же документов на языке XML, что и схема XSD:

```

EXAMPLES{joint-iso-itu-t asnl(1) examples(999) xml-defined-types(3)}
DEFINITIONS AUTOMATIC TAGS

XER INSTRUCTIONS ::=
BEGIN

IMPORTS String, Decimal, Int, Date, AnyType

FROM XSD
{joint-iso-itu-t asnl(1) specification(0) modules(0) xsd-module(1)};

EXAMPLES ::= SEQUENCE {
  personnelRecord      PersonnelRecord,
  number               Decimal,
  daysOfTheWeek        ListOfDays,
  namesOfMemberNations NamesOfMemberNations,
  fileIdentifier-list   [UNTAGGED]
    SEQUENCE (SIZE(1..MAX)) OF fileIdentifier FileIdentifier }

PersonnelRecord ::= [NAME AS UNCAPITALIZED] SEQUENCE {
  name      Name,
  title     XSD.String,
  number     INTEGER,
  dateOfHire Date,
  nameOfSpouse NameOfSpouse,
  children   Children }

NameOfSpouse ::= [NAME AS UNCAPITALIZED] Name

Name ::= [NAME AS UNCAPITALIZED] SEQUENCE {
  givenName   XSD.String,
  initial     XSD.String,
  familyName  XSD.String }

Children ::= [NAME AS UNCAPITALIZED] SEQUENCE {
  childInformation-list [UNTAGGED]
    SEQUENCE OF ChildInformation }

ChildInformation ::= SEQUENCE {
  name      Name,
  dateOfBirth Date }

```

```

ListOfDays ::= [LIST] SEQUENCE OF Day
Day ::=      ENUMERATED {monday, tuesday, wednesday, thursday, friday,
                        saturday, sunday}

NamesOfMemberNations ::= [NAME AS UNCAPITALIZED] [LIST] SEQUENCE OF XSD.String
FileIdentifier ::= [NAME AS UNCAPITALIZED] SEQUENCE {
    choice [UNTAGGED] CHOICE {
        serialNumber      Int,
        relativeName      XSD.String,
        unidentified      UNIDENTIFIED    } }

UNIDENTIFIED ::= [NAME AS LOWERCASED] XSD.AnyType

ENCODING-CONTROL XER
GLOBAL-DEFAULTS MODIFIED-ENCODINGS

END

```

C.3 Дальнейшие примеры

В этом подразделе во всех частичных примерах (примерах, не содержащих элемента **schema**) предполагается, что элементы языка XML, представляющие синтаксис XSD, находятся в контексте описания компоненты **default namespace**, компонента которой **namespace name** является компонентой **target namespace** схемы.

C.3.1 Схемные документы с информационными единицами элементов **import** и **include**

Приведенная ниже схема XSD составлена из двух пространств имен, которые составлены из четырех схемных файлов:

```

<!-- file "http://example.com/xyz/schema.xsd" -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xyz="http://example.com/xyz"
  targetNamespace="http://example.com/xyz">
  <xsd:element name="xyz-elem" type="xsd:string"/>
  <xsd:complexType name="Xyz-type">
    <xsd:attribute name="xyz-attr" type="xsd:boolean"/>
  </xsd:complexType>
</xsd:schema>

<!-- file "http://example.com/abc/main.xsd" -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:abc="http://example.com/abc"
  targetNamespace="http://example.com/xyz">
  <xsd:include schemaLocation="http://example.com/abc/sub1.xsd"/>
  <xsd:import namespace="http://www.w3.org/2001/XMLSchema"
    schemaLocation="http://example.com/xyz/schema.xsd"/>
  <xsd:redefine schemaLocation="http://example.com/abc/sub2.xsd">
    <xsd:attribute name="sub2-attr" type="xsd:token"/>
  </xsd:redefine>
  <xsd:element name="abc-elem" type="Xyz-type"/>
</xsd:schema>

<!-- file "http://example.com/abc/sub1.xsd" -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:abc="http://example.com/abc"
  targetNamespace="http://example.com/xyz">
  <xsd:element name="sub1-elem" type="xsd:string"/>
</xsd:schema>

<!-- file "http://example.com/abc/sub2.xsd" -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="sub2-elem" type="xsd:string"/>
  <xsd:attribute name="sub2-attr" type="xsd:string"/>
</xsd:schema>

```

Эти четыре схемных документа отображаются в двух следующих модулях ASN.1:

```

XYZ  -- Ссылка на модуль не стандартизирована
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

  Xyz-elem ::= [NAME AS UNCAPITALIZED] XSD.String
  Xyz-type ::= SEQUENCE {
    xyz-attr [ATTRIBUTE] BOOLEAN OPTIONAL }

  ENCODING-CONTROL XER
    GLOBAL-DEFAULTS MODIFIED-ENCODINGS
    GLOBAL-DEFAULTS CONTROL-NAMESPACE
    "http://www.w3.org/2001/XMLSchema-instance"
  END

ABC  -- Ссылка на модуль не стандартизирована
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
  IMPORTS
    Xyz-type FROM XYZ
    Token, String FROM XSD;

  Abc-elem ::= [NAME AS UNCAPITALIZED] Xyz-type
  Sub1-elem ::= [NAME AS UNCAPITALIZED] XSD.String
  Sub2-elem ::= [NAME AS UNCAPITALIZED] XSD.String
  Sub2-attr ::= [NAME AS UNCAPITALIZED] [ATTRIBUTE] XSD.Token

  ENCODING-CONTROL XER
    GLOBAL-DEFAULTS MODIFIED-ENCODINGS
    GLOBAL-DEFAULTS CONTROL-NAMESPACE
    "http://www.w3.org/2001/XMLSchema-instance"
  END

```

C.3.2 Отображение компонент simple type definition (определение простого типа)

C.3.2.1 Компонента simple type definition, полученная путем ограничения

См. полное множество примеров ограничений простого типа в примерах аспектов в подразделе C.3.3.

C.3.2.2 Компонента simple type definition, полученная с помощью списка

```

<xsd:simpleType name="Int-list">
  <xsd:list itemType="xsd:integer"/>
</xsd:simpleType>

<xsd:simpleType name="Int-10-to-100-list">
  <xsd:list>
    <xsd:simpleType>
      <xsd:restriction base="xsd:integer">
        <xsd:minInclusive value="10"/>
        <xsd:minInclusive value="100"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:list>
</xsd:simpleType>

```

Эти компоненты **simple type definition** отображаются в следующих присвоениях типов ASN.1:

```

Int-list ::= [LIST] SEQUENCE OF INTEGER
Int-10-to-100-list ::= [LIST] SEQUENCE OF INTEGER (10..100)

```

C.3.2.3 Компонента simple type definition, полученная путем объединения

```

<xsd:simpleType name="Int-or-boolean">
  <xsd:union itemType="xsd:integer xsd:boolean"/>
</xsd:simpleType>

<xsd:simpleType name="Time-or-int-or-boolean--or-dateRestriction">
  <xsd:union itemType="xsd:time Int-or-boolean">
    <xsd:simpleType>
      <xsd:restriction base="xsd:date">
        <xsd:minInclusive value="2003-01-01"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>

```

```

        </xsd:restriction>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>

```

Эти компоненты **simple type definition** отображаются в следующих присвоениях типов ASN.1:

```

Int-or-boolean ::= [USE-UNION] CHOICE {
  integer [NAMESPACE "http://www.w3.org/2001/XMLSchema"] INTEGER,
  boolean [NAMESPACE "http://www.w3.org/2001/XMLSchema"] BOOLEAN }

Time-or-int-or-boolean-or-dateRestriction ::= [USE-UNION] CHOICE {
  time [NAMESPACE "http://www.w3.org/2001/XMLSchema"] XSD.Time,
  integer [NAMESPACE "http://www.w3.org/2001/XMLSchema"] INTEGER,
  boolean [NAMESPACE "http://www.w3.org/2001/XMLSchema"] BOOLEAN,
  alt [NAME AS ""] XSD.Date (CONSTRAINED BY
    { /* minInclusive="2003-01-01" */ }) }

```

C.3.2.4 Отображение иерархий получения типов для компонент simple type definitions

```

<xsd:simpleType name="Int-10-to-50">
  <xsd:restriction base="xsd:integer">
    <xsd:minExclusive value="10"/>
    <xsd:maxExclusive value="50"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Ten-multiples">
  <xsd:restriction base="Int-10-to-50">
    <xsd:enumeration value="20"/>
    <xsd:enumeration value="30"/>
    <xsd:enumeration value="40"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Twenty-multiples">
  <xsd:restriction base="Ten-multiples">
    <xsd:pattern value=".*[02468]0|0"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="Stock-level">
  <xsd:extension base="Int-10-to-50">
    <xsd:attribute name="procurement" type="Int-10-to-50"/>
  </xsd:extension>
</xsd:complexType>

```

Эти компоненты **simple type definition** отображаются в следующих присвоениях типов ASN.1:

```

Int-10-to-50 ::= INTEGER (10<..<50)

Ten-multiples ::= ENUMERATED {int20(20), int30(30), int40(40)}

Twenty-multiples ::= ENUMERATED {int20(20), int40(40)}

Stock-level ::= SEQUENCE {
  procurement [ATTRIBUTE] Int-10-to-50 OPTIONAL,
  base Int-10-to-50-derivations }

Ten-multiples-derivations ::= [USE-TYPE] CHOICE {
  ten-multiples [NAME AS CAPITALIZED] Ten-multiples,
  twenty-multiples [NAME AS CAPITALIZED] Twenty-multiples }

Int-10-to-50-derivations ::= [USE-TYPE] CHOICE {
  int-10-to-50 [NAME AS CAPITALIZED] Int-10-to-50,
  ten-multiples [NAME AS CAPITALIZED] Ten-multiples,
  twenty-multiples [NAME AS CAPITALIZED] Twenty-multiples,
  stock-level [NAME AS CAPITALIZED] Stock-level }

```

C.3.3 Отображение компонент facet (аспектов)

C.3.3.1 Аспекты length, minLength и maxLength

```

<xsd:simpleType name="String-10">
  <xsd:restriction base="xsd:string">
    <xsd:length value="10"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<xsd:simpleType name="String-5-to-10">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="5"/>
    <xsd:maxLength value="10"/>
  </xsd:restriction>
</xsd:simpleType>

```

Эти две компоненты **simple type definition** отображаются в следующих присвоениях типов ASN.1:

```

String-10 ::= XSD.String (SIZE(10))
String-5-to-10 ::= XSD.String (SIZE(5..10))

```

C.3.3.2 Аспект pattern (комбинация)

```

<xsd:simpleType name="My-filename">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[&#00;-&#FF;]*"/>
    <xsd:pattern value="?([^\"]*)" />
  </xsd:restriction>
</xsd:simpleType>

```

Эта компонента **simple type definition** отображается в следующем присвоении типа ASN.1:

```

My-filename ::= XSD.String
  (CONSTRAINED BY
    { /* Представление на языке XML комбинации XSD
      "&#00;-&#FF;" | "? ([^\"]*)" */ })

```

C.3.3.3 Аспект whiteSpace

```

<xsd:simpleType name="My-String">
  <xsd:restriction base="xsd:string">
    <xsd:whitespace value="preserve"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="My-NormalizedString">
  <xsd:restriction base="xsd:string">
    <xsd:whitespace value="replace"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="My-TokenString">
  <xsd:restriction base="xsd:string">
    <xsd:whitespace value="collapse"/>
  </xsd:restriction>
</xsd:simpleType>

```

Эти компоненты **simple type definition** отображаются в следующих присвоениях типов ASN.1:

```

My-String ::= XSD.String
My-NormalizedString ::= [WHITESPACE REPLACE] XSD.String
  (FROM ({0, 0, 0, 32} .. {0, 16, 255, 255}))
My-TokenString ::= [WHITESPACE REPLACE] XSD.String
  (FROM ({0, 0, 0, 32} .. {0, 16, 255, 255}))
  (PATTERN "([^\ ]([^\ ]| [^\ ])*)?")

```

C.3.3.4 Аспекты minInclusive, minExclusive, maxInclusive и maxExclusive

```

<xsd:simpleType name="Int-10-to-100">
  <xsd:restriction base="xsd:integer">
    <xsd:minExclusive value="10"/>
    <xsd:maxInclusive value="100"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Pi-approximation">
  <xsd:restriction base="xsd:double">
    <xsd:minExclusive value="3.14159"/>
    <xsd:maxExclusive value="3.1416"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<xsd:simpleType name="Morning">
  <xsd:restriction base="xsd:time">
    <xsd:minInclusive value="00:00:00"/>
    <xsd:maxExclusive value="12:00:00"/>
  </xsd:restriction>
</xsd:simpleType>

```

Эти компоненты **simple type definition** отображаются в следующих присвоениях типов ASN.1:

```

Int-10-to-100 ::= INTEGER (10<..100)

Pi-approximation ::= XSD.Double (3.14159<..<3.1416)

Morning ::= XSD.Time (CONSTRAINED BY
  { /* minInclusive="00:00:00" maxExclusive="12:00:00" */ })

```

C.3.3.5 Аспекты totalDigits и fractionDigits

```

<xsd:simpleType name="RefundableExpenses">
  <xsd:restriction base="xsd:decimal">
    <xsd:totalDigits="5"/>
    <xsd:fractionDigits value="2"/>
  </xsd:restriction>
</xsd:simpleType>

```

Эта компонента **simple type definition** отображается в следующем присвоении типа ASN.1:

```

RefundableExpenses ::= XSD.Decimal (CONSTRAINED BY
  { /* totalDigits="5" fractionDigits="2" */ })

```

C.3.3.6 Аспект enumeration

```

<xsd:simpleType name="FarmAnimals">
  <xsd:restriction base="xsd:normalizedString">
    <xsd:enumeration value="Horse"/>
    <xsd:enumeration value="Bull"/>
    <xsd:enumeration value="Cow"/>
    <xsd:enumeration value="Pig"/>
    <xsd:enumeration value="Duck"/>
    <xsd:enumeration value="Goose"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PrimeNumbersBelow30">
  <xsd:restriction base="xsd:integer">
    <xsd:enumeration value="2"/>
    <xsd:enumeration value="3"/>
    <xsd:enumeration value="5"/>
    <xsd:enumeration value="7"/>
    <xsd:enumeration value="11"/>
    <xsd:enumeration value="13"/>
    <xsd:enumeration value="17"/>
    <xsd:enumeration value="19"/>
    <xsd:enumeration value="23"/>
    <xsd:enumeration value="29"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="X680-release">
  <xsd:restriction base="xsd:gYearMonth">
    <xsd:enumeration value="2002-07"/>
    <xsd:enumeration value="1997-12"/>
    <xsd:enumeration value="1994-07"/>
  </xsd:restriction>
</xsd:simpleType>

```

Эти компоненты **simple type definition** отображаются в следующих присвоениях типов ASN.1:

```

FarmAnimals ::= ENUMERATED {лошадь, бык, корова, свинья, утка, гусь}

PrimeNumbersBelow30 ::= [USE-NUMBER] ENUMERATED {int2(2), int3(3), int5(5),
  int7(7), int11(11), int13(13), int17(17), int19(19), int23(23), int29(29)}

X680-release ::= XSD.GYearMonth ("2002-07" | "1997-12" | "1994-07")

```

В управляющую секцию кодирования XER включена следующая команда кодирования:

```
TEXT FarmAnimals:ALL AS CAPITALIZED
```

С.3.3.7 Аспект enumeration в сочетании с другими аспектами

Приведенные ниже примеры основаны на наследовании аспектов, использующих ограничение некоторых типов, определение которых дано в подразделе С.3.3.6.

```
<xsd:simpleType name="FarmAnimals-subset">
  <xsd:restriction base="FarmAnimals">
    <xsd:minLength value="4"/>
    <xsd:pattern value="[^oe]*/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PrimeNumbersBelow30-subset">
  <xsd:restriction base="PrimeNumbersBelow30">
    <xsd:minExclusive value="5"/>
    <xsd:pattern value=".*[23].*/>
  </xsd:restriction>
</xsd:simpleType>
```

Эти компоненты **simple type definition** отображаются в следующих присвоениях типов ASN.1:

```
/* Лошадь и гусь не удовлетворяют аспекту pattern
   Корова и свинья не удовлетворяют аспекту minLength */
FarmAnimals-subset ::= ENUMERATED {bull, duck}

/* 2, 3 и 5 не удовлетворяют аспекту minExclusive
   2, 5, 7, 11, 17 и 19 не удовлетворяют аспекту pattern */
PrimeNumbersBelow30-subset ::= [USE-NUMBER] ENUMERATED {int13(13), int23(23),
                                                         int29(29)}
```

В управляющую секцию кодирования XER включена следующая команда кодирования:

```
TEXT FarmAnimals-subset:ALL AS CAPITALIZED
```

С.3.4 Отображение компонент element declarations

С.3.4.1 Компоненты element declarations, компонента которых type definition является определяемой пользователем компонентой simple type definition или complex type definition верхнего уровня

```
<xsd:element name="Forename" type="xsd:token"/>
<xsd:element name="File" type="My-filename"/>
<xsd:element name="Value" type="Int-10-to-50"/>
```

Эти компоненты **element declaration** отображаются в следующих присвоениях типов ASN.1:

```
Forename ::= XSD.Token
File ::= My-filename
Value ::= Int-10-to-50-derivations
```

ПРИМЕЧАНИЕ. – Тип **"My-filename"** и его отображение в ASN.1 определены в подразделе С.3.3.2; тип **"Int-10-to-50"** и его отображение в ASN.1 определены в подразделе С.3.2.4.

С.3.4.2 Компоненты element declarations, компонента которых type definition является анонимной компонентой simple type definition или complex type definition

```
<xsd:element name="maxOccurs">
  <xsd:simpleType>
    <xsd:union memberTypes="xsd:nonNegativeInteger">
      <xsd:simpleType>
        <xsd:restriction base="xsd:token">
          <xsd:enumeration name="unbounded"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="address">
  <xsd:complexType>
```

```

<xsd:sequence>
  <xsd:element name="line-1" type="xsd:token"/>
  <xsd:element name="line-2" type="xsd:token"/>
  <xsd:element name="city" type="xsd:token"/>
  <xsd:element name="state" type="xsd:token" minOccurs="0"/>
  <xsd:element name="zip" type="xsd:token"/>
</xsd:sequence>
<xsd:attribute name="country" type="xsd:token"/>
</xsd:complexType>
</xsd:element>

```

Эти компоненты **element declaration** отображаются в следующих присвоениях типов ASN.1:

```

MaxOccurs ::= [NAME AS UNCAPITALIZED] [USE-UNION] CHOICE {
  nonNegativeInteger [NAMESPACE AS "http://www.w3.org/2001/XMLSchema"]
    XSD.NonNegativeInteger,
  alt [NAME AS ""] ENUMERATED {unbounded} }

Address ::= [NAME AS UNCAPITALIZED] SEQUENCE {
  Country [ATTRIBUTE] XSD.Token OPTIONAL,
  line-1 XSD.Token,
  line-2 XSD.Token,
  city XSD.Token,
  state XSD.Token OPTIONAL,
  zip XSD.Token }

```

С.3.4.3 Компоненты **element declarations**, являющиеся головным элементом группы замены элементов

```

<xsd:element name="Tic" type="xsd:integer" abstract="true"/>
<xsd:element name="Tac" type="xsd:byte" substitutionGroup="Tic"/>
<xsd:element name="Toe" substitutionGroup="Tic"/>
<xsd:element name="Foo" type="xsd:date"/>
<xsd:element name="Bar" substitutionGroup="Foo"/>

```

Эти компоненты **element declaration** отображаются в:

```

Tac ::= INTEGER (-128..127)
Toe ::= INTEGER
Tic-group ::= [UNTAGGED] CHOICE {
  tac [NAME AS CAPITALIZED] Tac,
  toe [NAME AS CAPITALIZED] Toe }

Foo ::= XSD.Date
Bar ::= XSD.Date

Foo-group ::= [UNTAGGED] CHOICE {
  foo [NAME AS CAPITALIZED] Foo,
  bar [NAME AS CAPITALIZED] Bar }

```

С.3.4.4 Компоненты **element declaration** с компонентой **value constraint** (ограничение значения), которая является значением **default** (по умолчанию)

С.3.4.4.1 Ниже приводится компонента **element declaration** с анонимной компонентой **simple type definition**, не используемой как компонента **base type definition** любого типа.

```

<xsd:element name="Telephone" type="xsd:token" default="undefined"/>

```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```

Telephone ::= [DEFAULT-FOR-EMPTY "undefined"] XSD.Token

```

С.3.4.4.2 Ниже приводится компонента **element declaration** с анонимной компонентой **complex type definition** с простым содержанием, не используемой как компонента **base type definition** какого-либо типа.

```

<xsd:element name="InternationalTelephone" default="undefined">
  <xsd:simpleContent>
    <xsd:extension base="xsd:token">
      <xsd:attribute name="country-code" type="xsd:integer"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:element>

```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```
InternationalTelephone ::= [DEFAULT-FOR-EMPTY "undefined"] SEQUENCE {
    country-code [ATTRIBUTE] INTEGER OPTIONAL,
    base [UNTAGGED] XSD.Token }
```

С.3.4.4.3 Ниже приводится компонента **element declaration** с анонимной компонентой **complex type definition**. Компонента **complex type definition** имеет сложное содержание, обладает свойством **mixed** и незаполняемостью и не используется как компонента **base type definition** какого-либо типа.

```
<xsd:element name="Description" default="absent" mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="bold" type="xsd:string"/>
        <xsd:element name="italic" type="xsd:string"/>
    </xsd:choice>
</xsd:element>
```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```
Description ::= [EMBED-VALUES] [DEFAULT-FOR-EMPTY "absent"] SEQUENCE {
    embed-values SEQUENCE OF XSD.String,
    choice-list [UNTAGGED] SEQUENCE OF [UNTAGGED] CHOICE {
        bold XSD.String,
        italic XSD.String } } (CONSTRAINED BY
    /* Должна соответствовать Рек. МСЭ-Т X.693 / ИСО/МЭК 8825-4, раздел 25 */)
```

С.3.4.4.4 Компонента **type definition** в компоненте **element declaration** в приведенном ниже примере используется как компонента **base type definition** другого типа.

В этом примере используются типы XSD и ASN.1 примера в подразделе С.3.2.4.

```
<xsd:element name="Quantity" type="Int-10-to-50" default="20"/>
```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```
Quantity ::= Int-10-to-50-deriv-default-20
```

Если еще был генерирован какой-либо тип ASN.1, соответствующий **Int-10-to-50** со значением по умолчанию "20", то также генерируется следующий тип:

```
Int-10-to-50-deriv-default-20 ::= [USE-TYPE] CHOICE {
    int-10-to-50 [NAME AS CAPITALIZED] [DEFAULT-FOR-EMPTY 20]
        Int-10-to-50,
    ten-multiples [NAME AS CAPITALIZED] [DEFAULT-FOR-EMPTY int20]
        Ten-multiples,
    twenty-multiples [NAME AS CAPITALIZED] [DEFAULT-FOR-EMPTY int20]
        Twenty-multiples,
    stock-level [NAME AS CAPITALIZED] [DEFAULT-FOR-EMPTY 20]
        Stock-level }
```

С.3.4.5 Компонента **element declaration** с компонентой **value constraint**, являющейся значением со свойством **fixed** (фиксированная)

С.3.4.5.1 Ниже приводится компонента **element declaration** с анонимной компонентой **simple type definition**, которая не используется как компонента **base type definition** какого-либо типа.

```
<xsd:element name="UnknownTelephone" type="xsd:token" fixed="undefined"/>
```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```
UnknownTelephone ::= [DEFAULT-FOR-EMPTY "undefined"] XSD.Token ("undefined")
```

С.3.4.5.2 Ниже приводится компонента **element declaration** с анонимной компонентой **complex type definition**. Эта компонента **complex type definition** имеет простое содержание и не используется как компонента **base type definition** какого-либо типа.

```
<xsd:element name="UnknownInternationalTelephone" fixed="undefined">
    <xsd:simpleContent>
        <xsd:extension base="xsd:token">
            <xsd:attribute name="country-code" type="xsd:integer"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:element>
```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```
UnknownInternationalTelephone ::= [DEFAULT-FOR-EMPTY "undefined"] SEQUENCE {
    country-code [ATTRIBUTE] INTEGER OPTIONAL,
    base [UNTAGGED] XSD.Token }
    (WITH COMPONENTS {..., base ("undefined")})
```

С.3.4.5.3 Ниже приводится компонента **element declaration** с анонимной компонентой **complex type definition**. Компонента **complex type definition** имеет сложное содержание, обладает свойством **mixed** и незаполняемостью и не используется как компонента **base type definition** какого-либо типа.

```
<xsd:element name="UnknownDescription" fixed="absent" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="bold" type="xsd:string"/>
    <xsd:element name="italic" type="xsd:string"/>
  </xsd:choice>
</xsd:element>
```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```
UnknownDescription ::= [EMBED-VALUES] [DEFAULT-FOR-EMPTY "absent"] SEQUENCE {
  embed-values      SEQUENCE OF XSD.String,
  choice-list       [UNTAGGED] SEQUENCE OF [UNTAGGED] CHOICE {
    bold            XSD.String,
    italic          XSD.String } }
(CONSTRAINED BY
  { /* Должна соответствовать Рек. МСЭ-Т X.693 / ИСО/МЭК 8825-4,
    раздел 25 */ })
(WITH COMPONENTS {embed-values ({ "absent" }),
  choice-list (SIZE(0)) })
```

С.3.4.5.4 Компонента **type definition** следующей компоненты **element declaration** является компонентой **simple type definition**, используемой как компонента **base type definition** другого типа.

В данном примере используются типы XSD и ASN.1 из примера в подразделе С.3.2.4.

```
<xsd:element name="Quantity" type="Int-10-to-50" fixed="20"/>
```

Данная компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```
Quantity ::= Int-10-to-50-deriv-fixed-20
```

Если еще не генерирован какой-либо тип ASN.1, соответствующий **Int-10-to-50** с фиксированным значением "20", то также генерируется следующий тип:

```
Int-10-to-50-deriv-fixed-20 ::= [USE-TYPE] CHOICE {
  int-10-to-50      [NAME AS CAPITALIZED] [DEFAULT-FOR-EMPTY 20]
    Int-10-to-50,
  ten-multiples     [NAME AS CAPITALIZED] [DEFAULT-FOR-EMPTY int20]
    Ten-multiples,
  twenty-multiples  [NAME AS CAPITALIZED] [DEFAULT-FOR-EMPTY int20]
    Twenty-multiples,
  stock-level       [NAME AS CAPITALIZED] [DEFAULT-FOR-EMPTY 20]
    Stock-level }
(WITH COMPONENTS {
  int-10-to-50 (20),
  ten-multiples (int20),
  twenty-multiples (int20),
  stock-level (WITH COMPONENTS { ..., base (20) }) })
```

С.3.4.6 Компоненты **element declaration**, которые обладают свойством **nillable** (обнуляемые)

С.3.4.6.1 В следующем примере показана компонента **element declaration**, обладающая свойством **nillable** и компонента которой **type definition** является встроенным типом данных XSD.

```
<xsd:element name="Nillable-1" type="xsd:string" nillable="true"/>
```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```
Nillable-1 ::= [USE-NIL] SEQUENCE {
  content XSD.String OPTIONAL }
```

С.3.4.6.2 В следующем примере показана компонента **element declaration**, которая обладает свойством **nillable** и компонента которой **type definition** является анонимной компонентой **complex type definition**.

```
<xsd:element name="Nillable-2" nillable="true">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="a" type="xsd:string"/>
      <xsd:element name="b" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="b" type="xsd:boolean"/>
  </xsd:complexType>
</xsd:element>
```

Эта компонента **element declaration** отображается в следующем присвоении типа ASN.1:

```

Nillable-2 ::= [USE-NIL] SEQUENCE {
    b      [ATTRIBUTE] BOOLEAN OPTIONAL,
    content SEQUENCE {
        a      XSD.String,
        b      XSD.String } OPTIONAL }

```

С.3.4.6.3 В приведенном ниже примере показана компонента **element declaration**, обладающая свойством **nillable** и компонента которой **type definition** является компонентой **complex type definition** верхнего уровня.

```

<xsd:complexType name="Foo">
    <xsd:sequence>
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="b" type="xsd:boolean"/>
</xsd:complexType>

<xsd:element name="Nillable-3" type="Foo" nillable="true"/>

```

Эти схемные компоненты отображаются в следующих присвоениях типов ASN.1:

```

Foo ::= SEQUENCE {
    b      [ATTRIBUTE] BOOLEAN OPTIONAL,
    a      XSD.String,
    b-1    [NAME AS "b"] XSD.String }

Foo-nillable ::= [USE-NIL] SEQUENCE {
    b      [ATTRIBUTE] BOOLEAN OPTIONAL,
    content SEQUENCE {
        a      XSD.String,
        b      XSD.String } OPTIONAL }

Nillable-3 ::= Foo-nillable

```

С.3.4.6.4 В приведенном ниже примере показана компонента **element declaration**, обладающая свойством **nillable**, компонента которой **type definition** является компонентой **complex type definition** верхнего уровня, и которая используется как компонента **base type definition** другой компоненты **complex type definition**.

В дополнение к схемным компонентам, приведенным в подразделе С.3.4.6.3, определены следующие схемные компоненты:

```

<xsd:complexType name="Bar">
    <xsd:complexContent>
        <xsd:extension base="Foo">
            <xsd:sequence>
                <xsd:element name="z" type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="c" type="xsd:boolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:element name="Nillable-4" type="Foo" nillable="true"/>

```

В дополнение к типу Foo из подраздела С.3.4.6.3 генерируются следующие типы ASN.1:

```

Bar ::= SEQUENCE {
    b      [ATTRIBUTE] BOOLEAN OPTIONAL,
    c      [ATTRIBUTE] BOOLEAN OPTIONAL,
    a      XSD.String,
    b-1    [NAME AS "b"] XSD.String,
    z      XSD.String }

Foo-derivations ::= [USE-TYPE] CHOICE {
    foo    [NAME AS CAPITALIZED] Foo,
    bar    [NAME AS CAPITALIZED] Bar }

Foo-nillable ::= [USE-NIL] SEQUENCE {
    b      [ATTRIBUTE] BOOLEAN OPTIONAL,
    content SEQUENCE {
        a      XSD.String,
        b      XSD.String } OPTIONAL }

```

```

Bar-nillable ::= [USE-NIL] SEQUENCE {
    b      [ATTRIBUTE] BOOLEAN OPTIONAL,
    c      [ATTRIBUTE] BOOLEAN OPTIONAL,
    content SEQUENCE {
        a      XSD.String,
        b      XSD.String,
        z      XSD.String } OPTIONAL }

Foo-deriv-nillable ::= [USE-TYPE] CHOICE {
    foo      [NAME AS CAPITALIZED] Foo-nillable,
    bar      [NAME AS CAPITALIZED] Bar-nillable }

Nillable-4 ::= Foo-deriv-nillable

```

С.3.5 Отображение компонент **attribute use** (использование атрибута) и **attribute declaration** (описание атрибута)

С.3.5.1 Ниже приводится пример компоненты **attribute declaration**, компонента которой **type definition** является определяемой пользователем компоненты **simple type definition** верхнего уровня.

```
<xsd:attribute name="name" type="NCName"/>
```

Эта компонента **attribute declaration** отображается в следующем присвоении типа ASN.1:

```
Name ::= [NAME AS UNCAPITALIZED] [ATTRIBUTE] XSD.NCName
```

С.3.5.2 Ниже приводится пример компоненты **attribute declaration** верхнего уровня, компонента которой **type definition** является анонимной компонентой **simple type definition**.

```

<xsd:attribute name="form">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="qualified"/>
      <xsd:enumeration value="unqualified"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>

```

Эта компонента **attribute declaration** отображается в следующем присвоении типа ASN.1:

```
Form ::= [NAME AS UNCAPITALIZED] [ATTRIBUTE] ENUMERATED {qualified, unqualified}
```

С.3.5.3 Ниже приводится пример компоненты **attribute use** с компонентой **value constraint**, которая является значением со свойством **default**.

Компонента **attribute declaration** с компонентой **name** как "форма", на которую делаются ссылки в данном примере, определена в подразделе С.3.5.2.

```

<xsd:complexType name="element">
  <xsd:attribute name="name" type="xsd:NCName" default="NAME"/>
  <xsd:attribute ref="form" default="qualified"/>
</xsd:complexType>

```

Эта компонента **complex type definition** отображается в следующем присвоении типа ASN.1:

```

Element ::= [NAME AS UNCAPITALIZED] SEQUENCE {
    name [ATTRIBUTE] XSD.NCName DEFAULT "NAME",
    form [ATTRIBUTE] Form DEFAULT qualified }

```

С.3.5.4 В этом примере представлена компонента **attribute declaration** верхнего уровня с компонентой **value constraint**, являющейся значением со свойством **default**, и компонентой **attribute use** с этой компонентой **attribute declaration**.

```

<xsd:attribute name="minOccurs" type="xsd:nonNegativeInteger" default="1"/>
<xsd:attribute name="maxOccurs" default="1"/>
  <xsd:simpleType>
    <xsd:union memberTypes="xsd:nonNegativeInteger" >
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="unbounded"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>
</xsd:attribute>

```

```

<xsd:complexType name="Particle">
  <xsd:sequence>
    <xsd:element name="particle"/>
  </xsd:sequence>
  <xsd:attribute ref="minOccurs"/>
  <xsd:attribute ref="maxOccurs" default="unbounded"/>
</xsd:complexType>

```

Эти схемные компоненты отображаются в следующих присвоениях типов ASN.1:

```

MinOccurs ::= [ATTRIBUTE] [NAME AS UNCAPITALIZED] XSD.NonNegativeInteger

MaxOccurs ::= [ATTRIBUTE] [NAME AS UNCAPITALIZED] [USE-UNION] CHOICE {
  nonNegativeInteger      [NAMESPACE AS "http://www.w3.org/2001/XMLSchema"]
                           XSD.NonNegativeInteger,
  alt                     [NAME AS ""] ENUMERATED {unbounded} }

Particle ::= SEQUENCE {
  minOccurs      [ATTRIBUTE] MinOccurs DEFAULT 1,
  maxOccurs      [ATTRIBUTE] MaxOccurs DEFAULT alt : unbounded,
  particle       XSD.AnyType }

```

C.3.5.5 В этом примере представлена компонента **attribute use**, компонента которой **attribute declaration** имеет компоненту **target namespace**, не обладающую свойством **absent**.

```

<xsd:complexType name="Ack">
  <xsd:attribute name="number" type="xsd:integer" form="qualified" />
</xsd:complexType>

```

Эта компонента **complex type definition** отображается в следующем присвоении типа ASN.1:

```

Ack ::= SEQUENCE {
  number [NAMESPACE AS "http://targetnamespaceForExample"] [ATTRIBUTE]
        INTEGER OPTIONAL }

```

C.3.6 Отображение компонент **model group definition** (определение группы моделей)

C.3.6.1 Ниже приводится компонента **model group definition**, компонента которой **model group** имеет компоненту **compositor** для компоненты **sequence**.

```

<xsd:group name="mySequence">
  <xsd:sequence>
    <xsd:element name="a" type="xsd:string"/>
    <xsd:element name="b" type="xsd:boolean"/>
  </xsd:sequence>
</xsd:group>

```

Эта компонента **model group definition** отображается в следующем присвоении типа ASN.1:

```

MySequence ::= [UNTAGGED] SEQUENCE {
  a XSD.String,
  b BOOLEAN }

```

C.3.6.2 Ниже приводится компонента **model group definition**, компонента которой **model group** имеет компоненту **compositor** со свойством **all**.

```

<xsd:group name="myAll ">
  <xsd:all>
    <xsd:element name="a" type="xsd:string"/>
    <xsd:element name="b" type="xsd:boolean"/>
  </xsd:all>
</xsd:group>

```

Эта компонента **model group definition** не отображается в ASN.1. См. в подразделе C.3.8.3.1 пример отображения компоненты **complex type definition**, где компонента **model group** данной компоненты **model group definition** встречается в виде важнейшей компоненты **model group**.

C.3.6.3 Ниже приводится компонента **model group definition**, компонента которой **model group** имеет компоненту **compositor** со свойством **choice**.

```

<xsd:group name="myChoice">
  <xsd:choice>
    <xsd:element name="am" type="xsd:string"/>
    <xsd:element name="bm" type="xsd:boolean"/>
  </xsd:choice>
</xsd:group>

```

Эта компонента **model group definition** отображается в следующем присвоении типа ASN.1:

```

MyChoice ::= [UNTAGGED] CHOICE {
  am XSD.String,
  bm BOOLEAN }

```

С.3.7 Отображение компонент **particle** (частица)

В ряде примеров с компонентой **particle** используется компонента **model group definition** из подраздела С.3.6.3 и соответствующий ей тип ASN.1.

С.3.7.1 В приведенном ниже примере показаны компоненты **particle** в компоненте **model group** с компонентой **compositor** со свойством **sequence**.

```
<xsd:complexType name="ElementSequence">
  <xsd:sequence>
    <xsd:element name="elem1" type="xsd:boolean"/>
    <xsd:element name="elem2" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="elem3" type="xsd:boolean" minOccurs="2" maxOccurs="5"/>
    <xsd:element name="elem4" type="xsd:boolean" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="elem5" type="xsd:boolean" minOccurs="5" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ModelGroupSequence">
  <xsd:sequence>
    <xsd:group ref="myChoice"/>
    <xsd:choice>
      <xsd:element name="a" type="xsd:string"/>
      <xsd:element name="b" type="xsd:string"/>
    </xsd:choice>
    <xsd:sequence>
      <xsd:element name="c" type="xsd:string"/>
      <xsd:element name="d" type="xsd:string"/>
    </xsd:sequence>
    <xsd:choice minOccurs="3" maxOccurs="12">
      <xsd:element name="e" type="xsd:string"/>
      <xsd:element name="f" type="xsd:string"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```
ElementSequence ::= SEQUENCE {
    elem1          BOOLEAN,
    elem2          BOOLEAN OPTIONAL,
    elem3-list     [UNTAGGED] SEQUENCE (SIZE(2..5)) OF elem3 BOOLEAN,
    elem4-list     [UNTAGGED] SEQUENCE OF elem4 BOOLEAN,
    elem5-list     [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF elem5 BOOLEAN }

ModelGroupSequence ::= SEQUENCE {
    myChoice       MyChoice,
    choice         [UNTAGGED] CHOICE {
        a          XSD.String,
        b          XSD.String },
    sequence       [UNTAGGED] SEQUENCE {
        c          XSD.String,
        d          XSD.String },
    choice-list    [UNTAGGED] SEQUENCE (SIZE(3..12)) OF [UNTAGGED] CHOICE {
        e          XSD.String,
        f          XSD.String } }
```

С.3.7.2 В приведенном ниже примере показаны компоненты **particle** из компоненты **model group** с компонентой **compositor** со свойством **all**.

```
<xsd:complexType name="ElementAll">
  <xsd:all>
    <xsd:element name="elem1" type="xsd:boolean"/>
    <xsd:element name="elem2" type="xsd:boolean" minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

Эта компонента **complex type definition** отображается в следующем присвоении типа ASN.1:

```
ElementAll ::= [USE-ORDER] SEQUENCE {
    order SEQUENCE OF ENUMERATED {elem1, elem2},
    elem1 XSD.String,
    elem2 XSD.String OPTIONAL }
(CONSTRAINED BY
  { /* Должна соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4,
    раздел 35 */ })
```

C.3.7.3 В приведенном ниже примере показаны компоненты **particle** в компоненте **model group** с компонентой **compositor** со свойством **choice**.

```
<xsd:complexType name="ElementSequence">
  <xsd:choice>
    <xsd:element name="elem1" type="xsd:boolean"/>
    <xsd:element name="elem2" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="elem3" type="xsd:boolean" minOccurs="2" maxOccurs="5"/>
    <xsd:element name="elem4" type="xsd:boolean" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="elem5" type="xsd:boolean" minOccurs="5" maxOccurs="unbounded"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="ModelGroupChoice">
  <xsd:choice>
    <xsd:group ref="myChoice"/>
    <xsd:choice>
      <xsd:element name="a" type="xsd:string"/>
      <xsd:element name="b" type="xsd:string"/>
    </xsd:choice>
    <xsd:sequence>
      <xsd:element name="c" type="xsd:string"/>
      <xsd:element name="d" type="xsd:string"/>
    </xsd:sequence>
    <xsd:choice minOccurs="3" maxOccurs="12">
      <xsd:element name="e" type="xsd:string"/>
      <xsd:element name="f" type="xsd:string"/>
    </xsd:choice>
  </xsd:choice>
</xsd:complexType>
```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```
ElementSequence ::= SEQUENCE {
  choice [UNTAGGED] CHOICE {
    elem1      BOOLEAN,
    elem2-list [UNTAGGED] SEQUENCE (SIZE(0..1)) OF elem2 BOOLEAN,
    elem3-list [UNTAGGED] SEQUENCE (SIZE(2..5)) OF elem3 BOOLEAN,
    elem4-list [UNTAGGED] SEQUENCE OF elem4 BOOLEAN,
    elem5-list [UNTAGGED] SEQUENCE (SIZE(5..MAX)) OF elem5 BOOLEAN } }

ModelGroupChoice ::= SEQUENCE {
  choice [UNTAGGED] CHOICE {
    myChoice      MyChoice,
    choice [UNTAGGED] CHOICE {
      a          XSD.String,
      b          XSD.String },
    sequence [UNTAGGED] SEQUENCE {
      c          XSD.String,
      d          XSD.String }
    choice-list [UNTAGGED] SEQUENCE (SIZE(3..12)) OF [UNTAGGED] CHOICE {
      e          XSD.String,
      f          XSD.String } } }
```

C.3.8 Отображение компонент complex type definitions

C.3.8.1 В приведенном ниже примере показана компонента **complex type definition**, компонента которой **content type** обладает свойством **empty**.

```
<xsd:complexType name="Null"/>

<xsd:complexType name="Ack">
  <xsd:sequence/>
  <xsd:attribute name="packetNumber" type="xsd:integer"/>
</xsd:complexType>
```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```
Null ::= SEQUENCE {}

Ack ::= SEQUENCE {
  packetNumber [ATTRIBUTE] INTEGER OPTIONAL }
```

C.3.8.2 В приведенном ниже примере показана компонента **complex type definition**, компонента которой **content type** является компонентой **simple type definition**.

```
<xsd:complexType name="Formatted">
  <xsd:simpleContent>
    <xsd:extension base="xsd:token">
      <xsd:attribute name="format">
        <xsd:simpleType>
```

```

        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="bold"/>
            <xsd:enumeration value="italic"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

```

Эта компонента **complex type definition** отображается в следующем присвоении типа ASN.1:

```

Formatted ::= SEQUENCE {
    Format      [ATTRIBUTE] ENUMERATED {bold, italic} OPTIONAL,
    Base       [UNTAGGED] XSD.Token }

```

C.3.8.3 В следующих примерах представлены компоненты **complex type definition**, компонента которых **content type** является содержательной моделью со свойством **element-only** (только элементы).

C.3.8.3.1 В следующем примере компонента **content type** является компонентой **model group** компоненты **model group definition**.

В этом примере используются типы, определяемые в подразделе C.3.6.

```

<xsd:complexType name="MyComplexType-1">
    <xsd:group ref="myAll"/>
</xsd:complexType>

<xsd:complexType name="MyComplexType-2">
    <xsd:group ref="myChoice" />
</xsd:complexType>

<xsd:complexType name="MyComplexType-3">
    <xsd:group ref="mySequence" maxOccurs="100"/>
</xsd:complexType>

```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```

MyComplexType-1 ::= [USE-ORDER] SEQUENCE {
    order      SEQUENCE OF ENUMERATED {a,b},
    a          XSD.String,
    b BOOLEAN }
(CONSTRAINED BY
    { /* Должны соответствовать Рек. МСЭ-Т X.693 / ИСО/МЭК 8825-4,
      раздел 35 */ })

MyComplexType-2 ::= SEQUENCE {
    myChoice MyChoice }

MyComplexType-3 ::= SEQUENCE {
    mySequence-list SEQUENCE (SIZE(1..100)) OF MySequence }

```

C.3.8.3.2 В следующем примере компонента **content type** является компонентой **model group**, чья компонента **compositor** обладает свойством **choice**.

```

<xsd:complexType name="MyComplexType-4">
    <xsd:choice>
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="MyComplexType-5">
    <xsd:choice minOccurs="0">
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="MyComplexType-6">
    <xsd:choice maxOccurs="5">
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:choice>
</xsd:complexType>

```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```

MyComplexType-4 ::= SEQUENCE {
    choice [UNTAGGED] CHOICE {
        a XSD.String,
        b BOOLEAN } }

```

```

MyComplexType-5 ::= SEQUENCE {
    choice [UNTAGGED] CHOICE {
        a XSD.String,
        b BOOLEAN } OPTIONAL }

MyComplexType-6 ::= SEQUENCE {
    choice-list [UNTAGGED] SEQUENCE (SIZE(1..5)) OF [UNTAGGED] CHOICE {
        a XSD.String,
        b BOOLEAN } }

```

С.3.8.3.3 В следующем примере компонента **content type** является компонентой **model group**, компонента которой **compositor** обладает свойством **all**.

```

<xsd:complexType name="MyComplexType-7">
    <xsd:all>
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:all>
</xsd:complexType>

<xsd:complexType name="MyComplexType-8">
    <xsd:all minOccurs="0">
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:all>
</xsd:complexType>

```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```

MyComplexType-7 ::= [USE-ORDER] SEQUENCE {
    order      SEQUENCE OF ENUMERATED {a,b},
    a          XSD.String,
    b          BOOLEAN }
    (CONSTRAINED BY
        { /* Должны соответствовать Рек. МСЭ-Т X.693 / ИСО/МЭК 8825-4,
          раздел 35 */ })

MyComplexType-8 ::= [USE-ORDER] SEQUENCE {
    order      SEQUENCE OF ENUMERATED {a,b},
    a          XSD.String OPTIONAL,
    b          BOOLEAN OPTIONAL }
    (CONSTRAINED BY
        { /* Должны соответствовать Рек. МСЭ-Т X.693 / ИСО/МЭК 8825-4,
          раздел 35 */ })

```

С.3.8.3.4 В следующем примере компонента **content type** является компонентой **model group**, компонента которой **compositor** есть компонента **sequence**.

```

<xsd:complexType name="MyComplexType-9">
    <xsd:sequence>
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MyComplexType-10">
    <xsd:sequence minOccurs="0">
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MyComplexType-11">
    <xsd:sequence maxOccurs="5">
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>

```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```

MyComplexType-9 ::= SEQUENCE {
    a XSD.String,
    b BOOLEAN }

MyComplexType-10 ::= SEQUENCE {
    sequence [UNTAGGED] SEQUENCE {
        a XSD.String,
        b BOOLEAN } OPTIONAL }

```

```
MyComplexType-11 ::= SEQUENCE {
    sequence-list [UNTAGGED] SEQUENCE (SIZE(1..5)) OF [UNTAGGED] SEQUENCE {
        a XSD.String,
        b BOOLEAN } }

```

С.3.8.4 В следующем примере показана компонента **complex type definition**, компонента которой **content type** является содержательной моделью со свойством **mixed**.

```
<xsd:complexType name="MyComplexType-12" mixed="true">
    <xsd:sequence>
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MyComplexType-13" mixed="true">
    <xsd:all>
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:all>
</xsd:complexType>

<xsd:complexType name="MyComplexType-14" mixed="true">
    <xsd:choice>
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="MyComplexType-15" mixed="true">
    <xsd:all minOccurs="0">
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:all>
</xsd:complexType>

<xsd:complexType name="MyComplexType-16">
    <xsd:sequence maxOccurs="unbounded" minOccurs="0">
        <xsd:element name="a" type="xsd:string"/>
        <xsd:element name="b" type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>

```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```
MyComplexType-12 ::= [EMBED-VALUES] SEQUENCE {
    embed-values      SEQUENCE OF XSD.String,
    a                 XSD.String,
    b                 BOOLEAN }
    (CONSTRAINED BY
        { /* Должна соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4,
            раздел 25 */ })

MyComplexType-13 ::= [EMBED-VALUES] [USE-ORDER] SEQUENCE {
    embed-values      SEQUENCE OF XSD.String,
    order             SEQUENCE OF ENUMERATED {a,b},
    a                 XSD.String,
    b                 BOOLEAN }
    (CONSTRAINED BY
        { /* Должна соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4,
            раздел 25 */ })
    (CONSTRAINED BY
        { /* Должна соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4,
            раздел 35 */ })

MyComplexType-14 ::= [EMBED-VALUES] SEQUENCE {
    embed-values      SEQUENCE OF XSD.String,
    choice            [UNTAGGED] CHOICE {
        a             XSD.String,
        b             BOOLEAN } }
    (CONSTRAINED BY
        { /* Должна соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4,
            раздел 25 */ })

MyComplexType-15 ::= [EMBED-VALUES] [USE-ORDER] SEQUENCE {
    embed-values      SEQUENCE OF XSD.String,
    order             SEQUENCE OF ENUMERATED {a,b},
    a                 XSD.String OPTIONAL,
    b                 BOOLEAN OPTIONAL }

```

```

(CONSTRAINED BY
  { /* Должна соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4,
    раздел 35 */ })
(CONSTRAINED BY
  { /* Должна соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4,
    раздел 25 */ })

MyComplexType-16 ::= [EMBED-VALUES] SEQUENCE {
  embed-values      SEQUENCE OF XSD.String,
  sequence-list     [UNTAGGED] SEQUENCE OF [UNTAGGED] SEQUENCE {
    a                XSD.String,
    b                BOOLEAN } }
(CONSTRAINED BY
  { /* Должна соответствовать Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4,
    раздел 25 */ })

```

C.3.8.5 В следующем примере представлены компоненты **attribute uses** в компоненте **complex type definition**, образованной при использовании компоненты **attribute group definition**.

```

<xs:attributeGroup name="AG1">
  <xs:attribute name="a1" type="xs:string"/>
  <xs:attribute name="a2" type="xs:string"/>
  <xs:attribute name="a3" type="xs:decimal"/>
</xs:attributeGroup>

<xs:attributeGroup name="AG2">
  <xs:attribute name="a1" use="prohibited"/>
  <xs:attribute name="a3" type="xs:integer"/>
</xs:attributeGroup>

<xs:complexType name="MyComplexType-17">
  <xs:attribute name="a4" type="xs:boolean"/>
  <xs:attribute name="a5" type="xs:boolean"/>
  <xs:attributeGroup ref="AG1"/>
</xs:complexType>

<xs:complexType name="MyComplexType-18">
  <xs:complexContent>
    <xs:restriction base="MyComplexType-17">
      <xs:attributeGroup ref="AG2"/>
      <xs:attribute name="a4" use="prohibited"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```

MyComplexType-17 ::= SEQUENCE {
  a1 [ATTRIBUTE] XSD.String OPTIONAL,
  a2 [ATTRIBUTE] XSD.String OPTIONAL,
  a3 [ATTRIBUTE] XSD.Decimal OPTIONAL,
  a4 [ATTRIBUTE] BOOLEAN OPTIONAL,
  a5 [ATTRIBUTE] BOOLEAN OPTIONAL }

MyComplexType-18 ::= SEQUENCE {
  a2 [ATTRIBUTE] XSD.String OPTIONAL,
  a3 [ATTRIBUTE] INTEGER OPTIONAL,
  a5 [ATTRIBUTE] BOOLEAN OPTIONAL }

MyComplexType-17-derivations ::= [USE-TYPE] CHOICE {
  myComplexType-17 [NAME AS CAPITALIZED] MyComplexType-17,
  myComplexType-18 [NAME AS CAPITALIZED] MyComplexType-18 }

```

C.3.8.6 Получение компонент **complex type definition**.

```

<xsd:complexType name="MyComplexType-19">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="a" type="xsd:string"/>
    <xsd:element name="b" type="xsd:boolean"/>
    <xsd:element name="c" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="attr1" type="xsd:short" use="required"/>
  <xsd:attribute name="attr2" type="xsd:short"/>
</xsd:complexType>

<xsd:complexType name="MyComplexType-20">
  <xsd:complexContent>
    <xsd:restriction base="MyComplexType-19">
      <xsd:sequence>
        <xsd:element name="a" type="xsd:token"/>
        <xsd:element name="b" type="xsd:boolean"/>

```

```

        </xsd:sequence>
        <xsd:attribute name="attr2" type="xsd:short" use="prohibited"/>
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="MyComplexType-21">
    <xsd:complexContent>
        <xsd:extension base="MyComplexType-20">
            <xsd:sequence>
                <xsd:element name="d" type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="attr3" type="xsd:boolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```

MyComplexType-19 ::= SEQUENCE {
    attr1          [ATTRIBUTE] XSD.Short,
    attr2          [ATTRIBUTE] XSD.Short OPTIONAL,
    sequence-list  [UNTAGGED] SEQUENCE OF [UNTAGGED] SEQUENCE {
        a          XSD.String,
        b          BOOLEAN,
        c          BOOLEAN OPTIONAL } }

MyComplexType-20 ::= SEQUENCE {
    attr1          [ATTRIBUTE] XSD.Short,
    a              XSD.Token,
    b              BOOLEAN }

MyComplexType-21 ::= SEQUENCE {
    attr1          [ATTRIBUTE] XSD.Short,
    attr3          [ATTRIBUTE] BOOLEAN OPTIONAL,
    a              XSD.String,
    b              BOOLEAN,
    d              XSD.String }

MyComplexType-20-derivations ::= [USE-TYPE] CHOICE {
    myComplexType-20 [NAME AS CAPITALIZED] MyComplexType-20,
    myComplexType-21 [NAME AS CAPITALIZED] MyComplexType-21 }

MyComplexType-19-derivations ::= [USE-TYPE] CHOICE {
    myComplexType-19 [NAME AS CAPITALIZED] MyComplexType-19,
    myComplexType-20 [NAME AS CAPITALIZED] MyComplexType-20,
    myComplexType-21 [NAME AS CAPITALIZED] MyComplexType-21 }

```

C.3.9 Отображение атрибутов wildcard (метасимвол)

В этих примерах предполагается, что компонентой **target namespace** будет следующий URI: "http://www.asn1.org/X694/wildcard".

C.3.9.1 Атрибут wildcard.

```

<xsd:complexType name="AnyAttribute-1">
    <xsd:anyAttribute namespace="##any"/>
</xsd:complexType>

<xsd:complexType name="AnyAttribute-2">
    <xsd:anyAttribute namespace="##other"/>
</xsd:complexType>

<xsd:complexType name="AnyAttribute-3">
    <xsd:anyAttribute namespace="##targetNamespace"/>
</xsd:complexType>

<xsd:complexType name="AnyAttribute-4">
    <xsd:anyAttribute namespace="##local http://www.asn1.org/X694/attribute"/>
</xsd:complexType>

<xsd:complexType name="AnyAttribute-5">
    <xsd:complexContent>
        <xsd:extension base="AnyAttribute-4">
            <xsd:anyAttribute namespace="##targetNamespace"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```
AnyAttribute-1 ::= SEQUENCE {
    attr [ANY-ATTRIBUTES] SEQUENCE (CONSTRAINED BY {
        /* Каждый элемент должен отвечать формату "AnyAttributeFormat",
           описанному в Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */})
    OF XSD.String }

AnyAttribute-2 ::= SEQUENCE {
    attr [ANY-ATTRIBUTES EXCEPT "http://www.asn1.org/X694/wildcard"]
    SEQUENCE (CONSTRAINED BY {
        /* Каждый элемент должен отвечать формату "AnyAttributeFormat",
           описанному в Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */})
    OF XSD.String }

AnyAttribute-3 ::= SEQUENCE {
    attr [ANY-ATTRIBUTES FROM "http://www.asn1.org/X694/wildcard"]
    SEQUENCE (CONSTRAINED BY {
        /* Каждый элемент должен отвечать формату "AnyAttributeFormat",
           описанному в Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */})
    OF XSD.String }

AnyAttribute-4 ::= SEQUENCE {
    attr [ANY-ATTRIBUTES FROM ABSENT
        "http://www.asn1.org/X694/attribute"]
    SEQUENCE (CONSTRAINED BY {
        /* Каждый элемент должен отвечать формату "AnyAttributeFormat",
           описанному в Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */})
    OF XSD.String }

AnyAttribute-5 ::= SEQUENCE {
    attr [ANY-ATTRIBUTES FROM ABSENT
        "http://www.asn1.org/X694/attribute"
        "http://www.asn1.org/X694/wildcard"]
    SEQUENCE (CONSTRAINED BY {
        /* Каждый элемент должен отвечать формату "AnyAttributeFormat",
           описанному в Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */})
    OF XSD.String }
```

C.3.9.2 Ниже приводится пример атрибута **wildcard** содержательной модели.

```
<xsd:complexType name="Any-1">
    <xsd:sequence>
        <xsd:any namespace="##any"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Any-2">
    <xsd:sequence>
        <xsd:any minOccurs="0" namespace="##other"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Any-3">
    <xsd:sequence>
        <xsd:any minOccurs="0" maxOccurs="unbounded" namespace="##local"/>
    </xsd:sequence>
</xsd:complexType>
```

Эти компоненты **complex type definition** отображаются в следующих присвоениях типов ASN.1:

```
Any-1 ::= SEQUENCE {
    elem [ANY-ELEMENT] XSD.String (CONSTRAINED BY {
        /* Будет отвечать формату "AnyElementFormat", описанному в
           Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */}) }

Any-2 ::= SEQUENCE {
    elem [ANY-ELEMENT EXCEPT ABSENT
        "http://www.asn1.org/X694/wildcard"]
    XSD.String (CONSTRAINED BY {
        /* Должен отвечать формату "AnyElementFormat", описанному в
           Рек. МСЭ-Т X.693 | ИСО/МЭК 8825-4, раздел 18 */})
    OPTIONAL }
```

```
Any-3 ::= SEQUENCE {
    elem-list SEQUENCE OF elem
        [ANY-ELEMENT FROM ABSENT] XSD.String (CONSTRAINED BY {
            /* Должен отвечать формату "AnyElementFormat", описанному в
               Рек. МСЭ-Т X.693 / ИСО/МЭК 8825-4, раздел 18 */}) }
```

ПРИМЕЧАНИЕ. — См. дополнительные примеры вычисления элемента "NamespaceRestriction" в примерах по атрибутам wildcards в подразделе C.3.9.1.

Приложение D

Использование отображения для обеспечения двоичных кодирований для схемы на языке XML консорциума W3C

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В данном Приложении приводится описание использования отображения, приведенного в настоящей Рекомендации | Международном стандарте в сочетании со стандартизированными правилами кодирования ASN.1 для обеспечения канонического и компактного двоичного кодирования данных, определяемых схемой XSD.

D.1 Кодирование схем XSD

D.1.1 Схемы XSD могут быть отображены в определениях типов ASN.1, как указано в тексте настоящей Рекомендации | Международного стандарта, а затем тип верхнего уровня может быть закодирован при использовании любого из правил кодирования ASN.1, описанных в Рекомендации МСЭ-Т X.690 | ИСО/МЭК 8825-1, Рекомендации МСЭ-Т X.691 | ИСО/МЭК 8825-2 и Рекомендации МСЭ-Т X.693 | ИСО/МЭК 8825-4.

D.1.2 Каждый из этих видов кодирования имеет соответствующее значение идентификатора объекта, которое может быть использовано для определения кодирования при передаче. Способ, с помощью которого такая идентификация сообщается на декодер, не входит в сферу рассмотрения настоящей Рекомендации | Международного стандарта.

D.1.3 Когда схема XSD не пересылается получателю методом, описанным в подразделе D.3, способ, с помощью которого получатель получает эту схему, не входит в сферу рассмотрения настоящей Рекомендации | Международного стандарта.

D.2 Передача без использования схемы XSD для схем

D.2.1 В этом методе предполагается, что получателю известна схема XSD, используемая отправителем.

D.2.2 На рисунке D.1 показано, как использовать отображение, описанное в данной Рекомендации | Международном стандарте для кодирования документов на языке XML с помощью правил кодирования ASN.1.

D.2.3 Отправитель и получатель используют одну и ту же (фиксированную) схему XSD для генерирования схемы ASN.1, которая в свою очередь передается компилятору ASN.1 для генерирования таблицы кодирования по правилам BER, DER, PER или XER для документов на языке XML, отвечающих этой схеме XSD.

D.3 Передача с использованием схемы XSD для схем

D.3.1 Поскольку существует однозначно определяемая схема XSD для схем, можно выполнять действия в два этапа (см. рисунок D.2).

D.3.2 Отправитель и получатель организуют модуль ASN.1 и кодер/декодер из схемы XSD для схем.

D.3.3 На первом этапе отправитель кодирует схему XSD по правилам BER, DER или PER для документа и посылает закодированную схему получателю. Получатель декодирует эту схему и, используя отображение схемы XSD в ASN.1 и компилятор ASN.1, генерирует модуль ASN.1 и кодер/декодер для документов на языке XML, отвечающих этой схеме.

D.3.4 На втором этапе отправитель кодирует документ на языке XML по правилам BER, DER, PER или XER и посылает закодированный документ получателю.

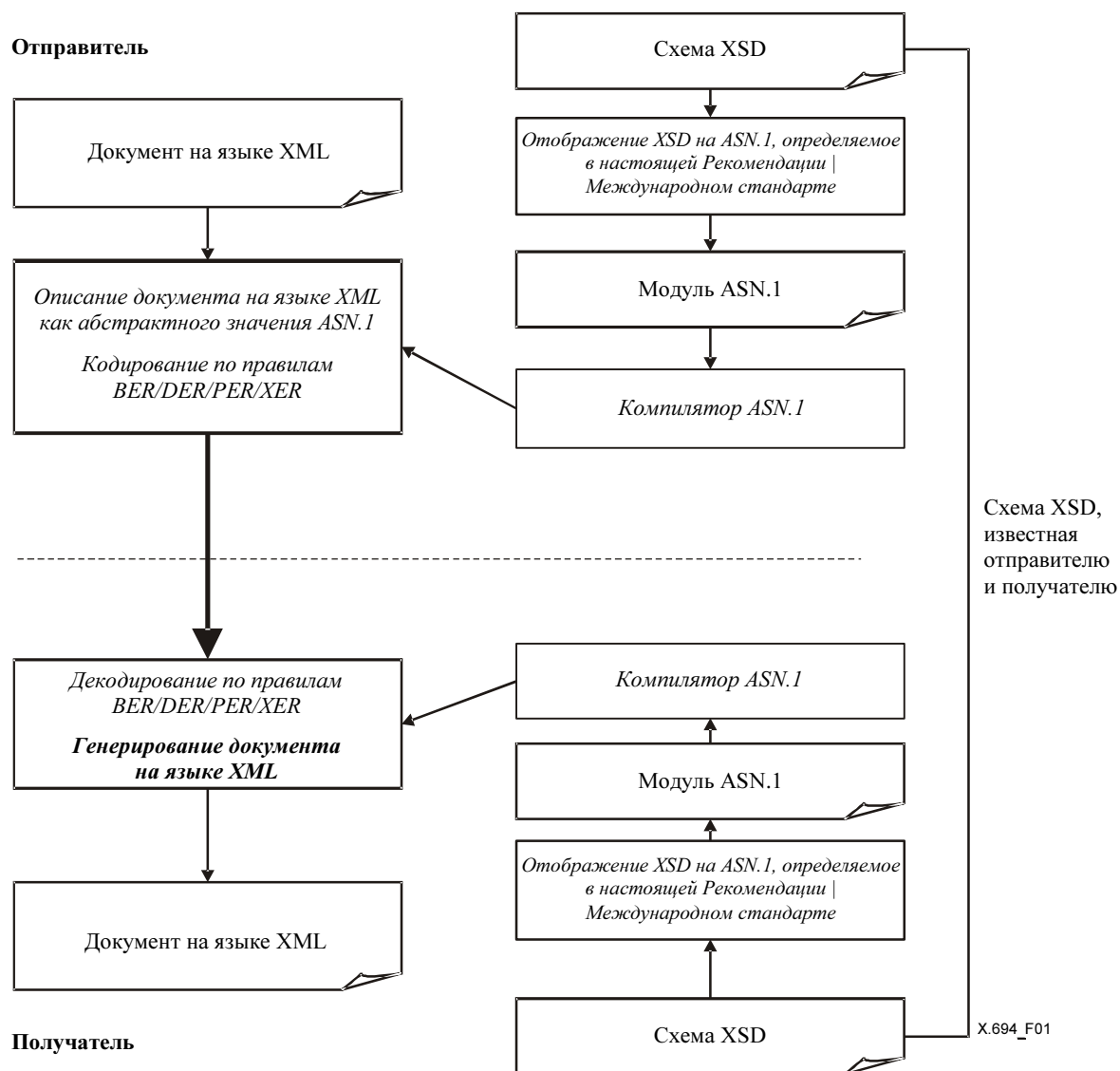


Рисунок D.1 – Передача документа на языке XML при использовании отображения XSD в ASN.1

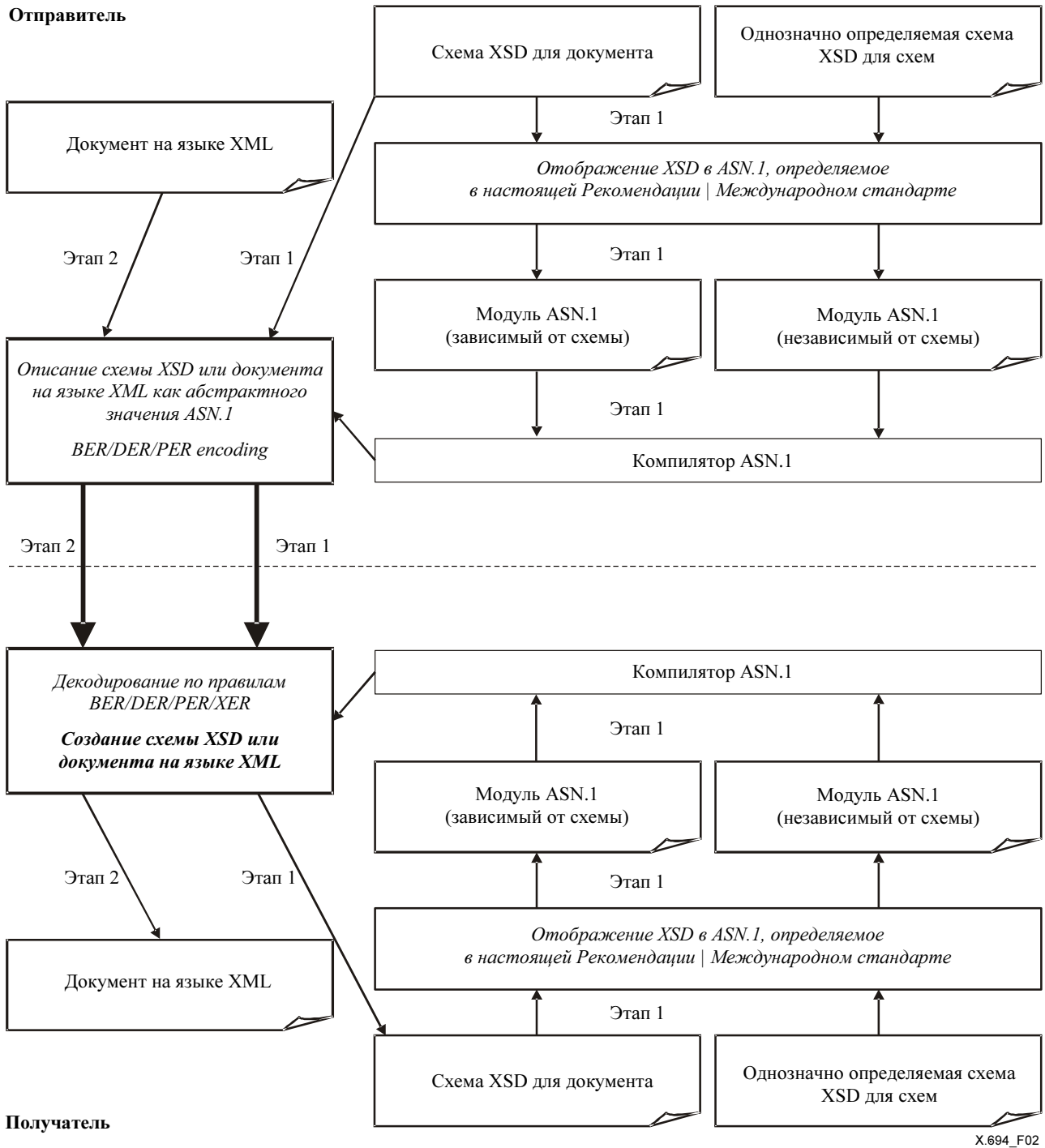


Рисунок D.2 – Передача схемы XSD и документа на языке XML при использовании отображения XSD в ASN.1

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия В	Средства выражения: определения, символы, классификация
Серия С	Общая статистика электросвязи
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	TMN и техническое обслуживание сетей: международные системы передачи, телефонные, телеграфные, факсимильные и арендованные каналы
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных и взаимосвязь открытых систем
Серия Y	Глобальная информационная инфраструктура, аспекты межсетевого протокола (IP) и сети следующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи