ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 8825-4— 2009

Информационная технология

ПРАВИЛА КОДИРОВАНИЯ АСН.1

Часть 4

Правила XML кодирования (XER)

ИСО/МЭК 8825-4:2002
Information technology — ACH.1 encoding rules.
Part 4: XML Encoding Rules (XER)
(IDT)

Издание официальное

Москва
Стандартинформ
2011

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0 — 2004 «Стандартизация в Российской Федерации. Основные положения»

Сведения о стандарте

1 РАЗРАБОТАН Государственным образовательным учреждением высшего профессионального образования Московским государственным технологическим университетом «Станкин» на основе собственного аутентичного перевода стандарта, указанного в разделе 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 461 «Информационно-коммуникационные технологии в образовании (ИКТО)»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 15 декабря 2009 г. № 969-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 8825-4:2002 «Информационная технология. Правила кодирования АСН.1. Часть 4. Правила XML кодирования (XER)» (ISO/IEC 8825-4:2002 «Information technology — ACH.1 encoding rules. Part 4: XML Encoding Rules (XER))», включая Изменение 1. EXTENDED-XER (Amendment 1: EXTENDED-XER).

Изменение к указанному международному стандарту, принятое после его официальной публикации, внесено в текст настоящего стандарта, а информация о его учете приведена в дополнительном приложении ГА.

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении ГБ

5 ВЗАМЕН ГОСТ Р ИСО/МЭК 8825-4:2006 Информационная технология. Правила кодирования АСН.1. Часть 4. Правила XML кодирования (XER)

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет.

© Стандартинформ, 2011

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения национального органа Российской Федерации по стандартизации.

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
2.1 Идентичные стандарты . . . 1
2.2 Дополнительные ссылки . . . 2
3 Определения . . . 2
3.1 Базовые правила кодирования ACH.1 (BER) . . . 2
3.2 Дополнительные определения . . . 3
4 Сокращения . . . 5
5 Правила кодирования . . . 5
6 Соответствие . . . 6
7 Базовые правила XML кодирования . . . 6
7.1 Продукционные правила полного BASIC-XER кодирования . . . 6
7.2 XML пролог . . . 6
7.3 Элемент XML документа . . . 7
7.4 Кодирование типа EXTERNAL . . . 7
7.5 Кодирование открытого типа . . . 7
7.6 Декодирование типов с маркерами расширения . . . 7
8 Канонические правила XML кодирования . . . 8
8.1 Общие правила для канонического XER . . . 8
8.2 Действительные значения . . . 8
8.3 Значение «битовая строка» . . . 8
8.4 Значение строки октетов (octetstring) . . . 8
8.5 Значение «последовательность» . . . 9
8.6 Значение «множество» . . . 9
8.7 Значение «множество-из» . . . 9
8.8 Значение «идентификатор объекта» . . . 9
8.9 Значение «относительный идентификатор объекта» . . . 9
8.10 Обобщенное время . . . 9
8.11 Универсальное синхронизированное время . . . 10
8.12 Значение открытого типа . . . 10
9 Правила расширенного XML кодирования . . . 10
9.1 Общие положения . . . 10
9.2 Соответствие EXTENDED-XER . . . 11
9.3 Структура результата расширенного XML кодирования . . . 13
10 Нотация, набор символов и лексические единицы, использованные в командах XML кодирования 14
11 Ключевые слова . . . 14
12 Назначение команды XML кодирования типу ACH.1 с использованием префикса типа . . . 15
13 Назначение команды кодирования XER с использованием управляющих разделов кодирования 19
13.1 Перечень назначений команд кодирования . . . 19
13.2 Идентификация целей для команды XML кодирования с использованием перечня целей . . 19
14 Множественное назначение команд кодирования XER . . . 25
14.1 Порядок, в котором рассматривается множественное назначение . . . 25
14.2 Эффект назначения команды кодирования отрицания . . . 25
14.3 Множественное назначение команд кодирования с множественными категориями . . . 26
14.4 Множественное назначение команд кодирования XER той же категории . . . 26
14.5 Разрешенные комбинации окончательных команд кодирования . . . 27
15 Поддержка команд кодирования XER для пространства имен и уточненных имен XML . . . 29
16 Спецификации кодирования EXTENDED-XER . . . 30
16.1 Элемент XML документа . . . 30
16.2 Продукционное правило “TypeNameOrModifiedTypeName” . . . 31
16.3 Продукционное правило “AttributeList” . . . 31
16.4 Продукционное правило “ExtendedXMLValue” . . . 31
16.5 Продукционное правило “ExtendedXMLChoiceValue” . . . 32

16.6 Продукционные правила “ExtendedXMLSequenceValue” и “ExtendedXMLSetValue” 33
16.7 Продукционные правила “ExtendedXMLSequenceOfValue” и “ExtendedXMLSetOfValue” . . . 34
16.8 Продукционное правило “ModifiedXMLIntegerValue” 35
16.9 Продукционное правило “ModifiedXMLRealValue” 35
17 Команда кодирования ANY-ATTRIBUTE 36
17.1 Общие положения 36
17.2 Ограничения 36
17.3 Действие на результат кодирования 37
18 Команда кодирования ANY-ELEMENT 38
18.1 Общие положения 38
18.2 Ограничения 38
18.3 Действие на результат кодирования 39
19 Команда кодирования ATTRIBUTE 39
19.1 Общие положения 39
19.2 Ограничения 39
19.3 Действие на результат кодирования 40
20 Команда кодирования BASE64 41
20.1 Общие положения 41
20.2 Ограничения 42
20.3 Действие на результат кодирования 42
21 Команда кодирования DECIMAL 42
21.1 Общие положения 42
21.2 Ограничения 43
21.3 Действие на результат кодирования 43
22 Команда кодирования DEFAULT-FOR-EMPTY 43
22.1 Общие положения 43
22.2 Ограничения 44
22.3 Действие на результат кодирования 45
23 Команда кодирования ELEMENT 46
23.1 Общие положения 46
23.2 Ограничения 46
23.3 Действие на результат кодирования 46
24 Команда кодирования EMBED-VALUES 46
24.1 Общие положения 46
24.2 Ограничения 46
24.3 Действие на результат кодирования 47
25 Команда кодирования GLOBAL-DEFAULTS 48
25.1 Общие положения 48
25.2 Ограничения 48
25.3 Действие на результат кодирования 48
26 Команда кодирования LIST 48
26.1 Общие положения 48
26.2 Ограничения 49
26.3 Действие на результат кодирования 49
27 Команда кодирования NAME 49
27.1 Общие положения 49
27.2 Ограничения 50
27.3 Действие на результат кодирования 51
28 Команда кодирования NAMESPACE 51
28.1 Общие положения 51
28.2 Ограничения 52
28.3 Действие на результат кодирования 52
29 Команда кодирования PI-OR-COMMENT 53
29.1 Общие положения 53
29.2 Ограничения 53
29.3 Действие на результат кодирования 53

30 Команда кодирования TEXT . . . 54
30.1 Общие положения . . . 54
30.2 Ограничения . . . 54
30.3 Действие на результат кодирования . . . 55
31 Команда кодирования UNTAGGED . . . 56
31.1 Общие положения . . . 56
31.2 Ограничения . . . 56
31.3 Действие на результат кодирования . . . 57
32 Команда кодирования USE-NIL . . . 57
32.1 Общие положения . . . 57
32.2 Ограничения . . . 58
32.3 Действие на результат кодирования . . . 58
33 Команда кодирования USE-NUMBER . . . 59
33.1 Общие положения . . . 59
33.2 Ограничения . . . 59
33.3 Действие на результат кодирования . . . 59
34 Команда кодирования USE-ORDER . . . 59
34.1 Общие положения . . . 59
34.2 Ограничения . . . 60
34.3 Действие на результат кодирования . . . 61
35 Команда кодирования USE-QNAME . . . 61
35.1 Общие положения . . . 61
35.2 Ограничения . . . 61
35.3 Действие на результат кодирования . . . 62
36 Команда кодирования USE-TYPE . . . 62
36.1 Общие положения . . . 62
36.2 Ограничения . . . 62
36.3 Действие на результат кодирования . . . 62
37 Команда кодирования USE-UNION . . . 63
37.1 Общие положения . . . 63
37.2 Ограничения . . . 63
37.3 Действие на результат кодирования . . . 64
38 Команда кодирования WHITESPACE . . . 65
38.1 Общие положения . . . 65
38.2 Ограничения . . . 65
38.3 Действие на результат кодирования . . . 65
39 Значения идентификатора объекта, ссылающиеся на правила кодирования . . . 66
Приложение А (справочное) Примеры кодирования BASIC-XER и CXER . . . 67
Приложение Б (справочное) Частичное XML содержание и детерминированные кодирования . . . 69
Приложение В (справочное) Примеры EXTENDED-XER кодирования с использованием команд кодирования XER . . . 72
Приложение ГА (справочное) Сведения об учете в тексте стандарта Изменения 1.EXTENDED-XER . 75
Приложение ГБ (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации . . . 85

Введение

Перспективы создания информационного общества и масштабы развития единой образовательной информационной среды (ЕОИС) обусловили необходимость разработки комплекса национальных стандартов по информационно-коммуникационным технологиям в образовании (ИКТО). Разработку комплекса национальных стандартов по ИКТО осуществляет ТК 461 по шести направлениям:

- общесистемные и основополагающие нормативные документы по стандартизации ИКТО;
- взаимосвязь открытых систем в образовании;
- автоматизированные информационные системы управления отраслью и образовательными учреждениями;
- образовательные среды и информационные ресурсы;
- обеспечение функциональной безопасности ИКТО;
- обеспечение качества электронного обучения.

Разрабатываемый ТК 461 комплекс национальных стандартов соответствует международным стандартам в области менеджмента качества (ISO TC 176) и информационных технологий (ИСО/МЭК JTC1), в том числе:

- системной и программной инженерии (ИСО/МЭК JTC1 SC7);
- телекоммуникаций и информационного обмена между системами(ИСО/МЭК JTC1 SC6);
- управления и обмена данными (ИСО/МЭК JTC1 SC32);
- описания документов и языков обработки (ИСО/МЭК JTC1 SC34);
- информационных технологий в обучении, образовании и подготовке (ISO/IES JTC1 SC36).

Данный национальный стандарт аутентичен международному стандарту ИСО/МЭК 8825-4:2002, устанавливающему правила кодирования, которые могут быть применены к значениям типов АСН.1, принятым в нотации, определенной в ИСО/МЭК 8824-1 и ИСО/МЭК 8824-2. Международные стандарты ИСО/МЭК 8824 (части 1-4) и ИСО/МЭК 8825 (части 1-5) определяют требования к информационному обмену между равноправными приложениями. На их основе должны разрабатываться требования к информационным средам и системам для образования, науки, госсектора, промышленности и других сфер деятельности, они также включают аутентичный перевод Изменения 1. EXTENDED-XER к указанному стандарту.

Абстрактная Синтаксическая Нотация версии 1 (АСН.1), используемая для формирования сообщений при информационном обмене между равноправными приложениями, определяется рядом национальных стандартов: ИСО/МЭК 8824-1, ИСО/МЭК 8824-2, ИСО/МЭК 8824-3, ИСО/МЭК 8824-4.

При использовании приведенных в данных стандартах правил кодирования создается синтаксис передачи сообщений. В спецификации этих правил кодирования подразумевается, что они также могут использоваться для декодирования.

Настоящий стандарт определяет правила кодирования, которые могут быть применены к значениям типов АСН.1, определенных с использованием нотации, спецификация которой приведена в указанных выше стандартах. Применение этих правил кодирования продуцирует синтаксис перехода для таких значений.

Существует несколько наборов правил кодирования, которые могут быть применены к значениям типов АСН.1. Настоящий стандарт определяет три набора правил кодирования, которые используют Расширяемый Язык Разметки (XML). На основе любого из этих правил кодирования создаются XML документы, соответствующие Рекомендации W3C XML 1.0. Первый набор называется «Базовые правила XML кодирования (BASIC-XER)». Второй набор называется «Канонические правила XML кодирования (CANONICAL-XER или CXER)», так как существует только один способ кодирования значений АСН.1 с использованием этих правил. (Канонические правила кодирования обычно применяются для приложений, имеющих особенности, связанные с безопасностью, например, использующие цифровую подпись). Третий набор называется «Расширенные правила XML кодирования (EXTENDED-XER)», допускающие дополнительные опции кодера и учитывающие команды кодирования, которые определяют вариации Базовых правил кодирования (BASIC-XER), чтобы осуществить поддержку специфических стилей XML документов (см. ниже). Эти «Расширенные правила кодирования» не являются каноническими.

Существует много аспектов XML представления данных (таких как использование XML атрибутов вместо дочерних элементов или перечней пробелов с разделителями), использование которых определяется стилем и выбором XML дизайнера. Если тип, определенный в спецификации АСН.1, закодирован по правилам BASIC-XER или CXER, то для XML представления применяется единственный фиксированный стиль без управления стилистическими особенностями пользователем. Настоящий стандарт специфициру-

ет синтаксис и семантику команд XER кодирования, которые определяют стилистические особенности XML в EXTENDED-XER кодировании. Команды XER кодирования могут также быть использованы для определения возможного включения команд XML обработки в EXTENDED-XER кодирование. Команды XER кодирования игнорируются правилами BASIC-XER и CXER, но используются EXTENDED-XER.

Примечание — «Стилистические особенности», такие как атрибуты или перечни пробелов с разделителями, могут также влиять на размер закодированного материала и сложность его обработки, поэтому использование таких особенностей это не только вопрос стиля. Если такие вопросы существенны, можно отдать предпочтение EXTENDED-XER перед BASIC-XER или CXER.

В разделах 1 — 4 приводятся описание области применения стандарта, нормативные ссылки, определения терминов и сокращения.

Раздел 5 содержит перечень наборов правил кодирования, определяемых в данном стандарте.

В разделе 6 указывается, в каких последующих разделах специфицируется динамическое соответствие различных правил кодирования, декларируется статическое соответствие, а также разъясняются варианты разрешения (запрещения) альтернативного кодирования.

В разделе 7 специфицируется BASIC-XER кодирование типов ACH.1.

В разделе 8 специфицируется CXER кодирование типов ACH.1.

В разделе 9 специфицируется EXTENDED-XER кодирование типов ACH.1, с использованием ссылок на более поздние разделы, которые определяют команды XER кодирования.

В разделах с 10-го по 13-й перечисляются и классифицируются команды XER кодирования и специфицируется синтаксис для их назначения типам или компонентам ACH.1, с использованием либо префикс типа XER (см. ИСО/МЭК 8824-1, 30.3), либо управляющей секции XER кодирования.

В разделе 14 определяется порядок предшествования, если команды XER кодирования присутствуют и в префиксе типа XER, и в управляющей секции XER кодирования.

В разделе 15 специфицируется поддержка команд XER кодирования для пространства имен XML при использовании EXTENDED-XER.

В разделе 16 специфицируется EXTENDED-XER кодирование.

В разделах с 17 по 39 специфицируются:

а) синтаксис каждой команды XER кодирования, используемой в префиксе типа или управляющей секции кодирования;

б) ограничения команд XER кодирования, которые могут быть ассоциированы с конкретным типом ACH.1 (следствие наследования и множественных назначений);

в) модификации правил XER кодирования, которые необходимы в EXTENDED-XER кодировании, когда применяются команды XER кодирования.

Приложение А информативно и содержит примеры BASIC-XER и CXER кодирования.

Приложение Б информативно и содержит описание частичного XML контента, созданного для таких конструкций, как последовательность и последовательность-из, которые утратили свои окружающие теги вместе с ограничениями спецификаций, облегчающими определение компонента ACH.1, связанного с элементом XML.

Приложение В информативно и содержит примеры команд XER кодирования и соответствующих результатов EXTENDED-XER кодирования.

Приложение ГА справочное и содержит сведения об учете в тексте стандарта Изменения 1. EXTENDED-XER.

Приложение ГБ справочное и содержит сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации.

*** Данное введение не является частью настоящего стандарта**

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ПРАВИЛА КОДИРОВАНИЯ АСН.1

Часть 4

Правила XML кодирования (XER)

Information technology. ACH.1 encoding rules. Part 4. XML encoding rules (XER)

Дата введения — 2011 — 01 — 01

1 Область применения

Настоящий стандарт устанавливает набор базовых правил XML кодирования (BASIC-XER), которые могут быть использованы с целью получения синтаксиса перехода для значений типов, определенных в ИСО/МЭК 8824-1 и ИСО/МЭК 8824-2. Настоящий стандарт определяет также набор канонических правил XML кодирования (CXER), которые предусматривают ограничения на базовые правила XML кодирования и позволяют производить уникальное кодирование для любого данного значения АСН.1. Настоящий стандарт специфицирует набор расширенных правил XML кодирования (EXTENDED-XER), добавляющий дополнительные опции кодера и позволяющий спецификатору АСН.1 варьировать кодирование, которое должно быть произведено в соответствии с BASIC-XER. Спецификация данных правил кодирования предусматривает возможность использовать их для декодирования.

Правила кодирования, специфицированные в настоящем стандарте:

- используются во время связи;
- предназначены для использования в тех случаях, когда отображение и/или обработка значений при помощи доступных XML средств (типа браузеров) являются определяющими в выборе правил кодирования;
- позволяют расширить абстрактный синтаксис путем добавления дополнительных значений для всех форм расширяемости, описанных в ИСО/МЭК 8824-1.

Настоящий стандарт устанавливает также синтаксис и семантику команд XER кодирования и правила для их назначения и сочетания. Команды XER кодирования могут быть использованы для управления расширенными правилами XML кодирования (EXTENDED-XER) для конкретных типов АСН.1.

2 Нормативные ссылки

Настоящий стандарт содержит положения, имеющие ссылки на нижеследующие рекомендации и международные стандарты, которые действовали во время издания. Все рекомендации и международные стандарты допускаются к пересмотру, и пользователи имеют возможность применять самые последние издания рекомендаций и стандартов. Члены МЭК и ИСО утверждают реестр действующих международных стандартов.

Бюро телекоммуникационной стандартизации, международного союза по телекоммуникациям поддерживает список действующих в настоящее время рекомендаций МСЭ-Т.

2.1 Идентичные стандарты

ИСО/МЭК 8824-1:2002, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1): Спецификация базовой нотации.

Издание официальное

ИСО/МЭК 8824-1:2002/Amd.1:2003, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1): Спецификация базовой нотации — Изменение 1: Поддержка расширенных правил XML кодирования (EXTENDED-XER).

ИСО/МЭК 8824-2:2002, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1): Спецификация информационного объекта.

ИСО/МЭК 8824-2:2002/Amd.1:2003, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1) — Изменение 1: Спецификация информационного объекта — Поддержка расширенных правил XML кодирования (EXTENDED-XER).

ИСО/МЭК 8824-3:2002, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1): Спецификация ограничений.

ИСО/МЭК 8824-4:2002, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1): Параметризация спецификаций АСН.1.

ИСО/МЭК 8825-1:2002, Информационные технологии — Правила кодирования АСН.1: Спецификация Базовых правил кодирования (BER), Канонических правил кодирования (CER) и Отличительных правил кодирования (DER).

ИСО/МЭК 8825-1:2002/Amd.1:2003, Информационные технологии — Правила кодирования АСН.1: Спецификация базовых правил кодирования (BER), Канонических правил кодирования (CER) и Отличительных правил кодирования (DER) — Изменение 1: Поддержка Расширенных правил XML кодирования (EXTENDED-XER).

ИСО/МЭК 8825-2:2002, Информационные технологии — Правила кодирования АСН.1: Спецификация Правил уплотненного кодирования (PER).

ИСО/МЭК 8825-2:2002/Amd.1:2003, Информационные технологии — Правила кодирования АСН.1: Спецификация Правил уплотненного кодирования (PER) — Изменение 1: Поддержка Расширенных правил XML кодирования (EXTENDED-XER).

ИСО/МЭК 8825-3:2002, Информационные технологии — Правила кодирования АСН.1: Спецификация Управляющей нотации кодирования (ECN).

2.2 Дополнительные ссылки

IETF RFC 2045 (1996) Многоцелевые расширения Интернет-почты (MIME). Часть 1: Формат Совокупности Интернет-сообщений.

IETF RFC 2141 (1997) Синтаксис Унифицированного имени ресурса (URN).

IETF RFC 2396 (1998) Унифицированные идентификаторы ресурса (URI): Типовой синтаксис.

IETF RFC 3061 (2001), Пространство имен URN объектных идентификаторов.

ИСО/МЭК 10646-1:2000, Информационные технологии — Универсальный многооктетный набор кодированных символов (UCS) — Часть 1: Архитектура и базовая многоязычная матрица.

The Unicode Standard, Version 3.2.0, Консорциум Юникода.

Примечание — Графические символы (и их коды), определенные в вышеуказанном стандарте идентичны тем, которые определены стандартом ИСО/МЭК 10646-1, но вышеуказанный стандарт включен в перечень, поскольку он также специфицирует имена управляющих символов.

W3C XML 1.0:2000, Расширяемый Язык Разметки (XML) 1.0 (Второе издание), Рекомендация W3C, Copyright © [6 октября 2000 г.] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University),

gost.e-ecolog.ru

W3C XML Namespaces:1999, Пространство имен в XML, Рекомендация W3C, Copyright © [14 января 1999 г.] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University),

gost.e-ecolog.ru

Примечание — Ссылка на документ из настоящего стандарта не дает ему, как автономному документу, статус международного стандарта.

3 Определения

Для настоящего стандарта применимы определения ИСО/МЭК 8824-1, а также следующие определения.

3.1 Базовые правила кодирования АСН.1 (BER)

В настоящем стандарте применяют следующие термины по ИСО/МЭК 8825-1:

а) значение данных;

б) динамическое соответствие;

в) кодирование (значений данных);
г) получатель;
д) отправитель;
е) статическое соответствие.

3.2 Дополнительные определения

В настоящем стандарте применяют следующие дополнительные термины с соответствующими определениями.

3.2.1 **ACH.1 схема** (ACH.1 schema): Описание содержания и структуры данных, использующее определение типов ACH.1.

Примечание — ACH.1 схема дает возможность применять правила кодирования для двоичного кодирования значений типа ACH.1 или кодирование, использующее XML.

3.2.1.1 **связанный тег пустого элемента** (associated empty-element tag): Тег пустого элемента XML, который может заменить предшествующий тег и связанный последующий тег, если таковые присутствуют.

Примечание — Некоторые команды кодирования удаляют связанные теги "XMLValue".

3.2.1.2 **связанные команды кодирования (для типа)** (associated encoding instructions (for a type)): Набор инструкций XER кодирования, связанных с типом.

3.2.1.3 **связанный последующий тег** (associated following tag): Концевой тег XML, следующий за "XMLValue" типа, в отсутствие команды кодирования, которые удаляют связанные теги.

3.2.1.4 **связанный предшествующий тег** (associated preceding tag): Начальный тег XML, предшествующий "XMLValue" типа, в отсутствие команды кодирования, которые удаляют связанные теги.

3.2.2 **канонический действительный XML документ (для ACH.1 схемы)** (canonical valid XML document (for an ACH.1 schema)): XML документ, который правильно построен (см. W3C XML 1.0) и чье содержание соответствует спецификации CXER для кодирования типа ACH.1, определенного ACH.1 схемой.

3.2.2.1 **посимвольно кодируемый тип** (character-encodable type): Тип ACH.1, к которому может быть применена команда кодирования ATTRIBUTE (см. 19.2.1).

3.2.2.2 **управляющее пространство имен** (control namespace): Пространство имен, используемое для идентификации атрибутов, которые выполняют функции или переносят значения, предназначенные для управления кодированием в соответствии с расширенными правилами EXTENDED-XER.

Примечание 1 — Примером может быть атрибут идентификации типа. Управляющее пространство имен по умолчанию приравнивается к пространству имен ACH.1, определенному в 16.9, но может быть изменено командой кодирования GLOBAL-DEFAULTS.

Примечание 2 — Управляющее пространство имен может также содержать имена для атрибутов, которые могут присутствовать, но которые игнорируются декодерами EXTENDED-XER (см. 9.2.10). Примером такого имени атрибута может быть schemaLocation.

3.2.2.3 **включенный тип ACH.1** (enclosed (ACH.1) type): Тип ACH.1, чье XML значение — "XMLValue" при кодировании в соответствии с базовыми правилами кодирования (BASIC-XER) включено непосредственно в "XMLValue" типа ACH.1 (включающий тип).

Примечание — Все типы в BASIC-XER и EXTENDED-XER кодировании являются включенными типами, если только они не используются как корневые типы (см. 9.3.1 б) в коде.

3.2.2.4 **включающий элемент (типа ACH.1)** (enclosing element (of an ACH.1 type)): Элемент "ExtendedXMLTypedValue", "ExtendedXMLChoiceValue", "ExtendedXMLNamedValue" или "ExtendedXMLDelimitedItem", который имеет "ExtendedXMLValue", кодирующий тип, в качестве своего "ExtendedXMLValue" (см. 16.1, 16.5, 16.6 и 16.7).

3.2.2.5 **включающий тип (типа ACH.1)** (enclosing type (of an ACH.1 type)): Тип ACH.1, чье XML значение — "XMLValue" при кодировании в соответствии с базовыми правилами кодирования (BASIC-XER) включает непосредственно "XMLValue" типа ACH.1 (включенный тип).

Примечание — Включающий тип может быть типом последовательности, типом множества, типом выбора, типом последовательности-из, типом множества-из, открытым типом, типом строки октетов или строки битов (с CONTAINING и без ENCODED BY).

3.2.2.6 **окончательная команда кодирования (для типа)** (final encoding instructions (for a type)): Набор команд XER кодирования, связанных с типом как результат законченной спецификации ACH.1 и применямый в продуцировании кода этого типа.

3.2.2.7 **унаследованные команды кодирования** (inherited encoding instructions): Команды XER кодирования, которые связаны с типом, идентифицируемым ссылкой на тип.

3.2.2.8 **имя в пространстве имен ограниченного использования** (namespace-qualified name): Имя в XML документе, которое имеет префикс пространства имен XML или является именем XML элемента в пределах объявленного пространства имен XML по умолчанию.

Примечание — Декларации пространства имен XML по умолчанию действуют только на имена элементов XML, но не на имена атрибутов. Префикс пространства имен может быть применен как к именам элементов, так и атрибутов.

3.2.2.9 **атрибут идентификации nil (пустое множество)** (nil identification attribute): Атрибут XML, который может появляться в любом элементе с целью идентификации контента с значением nil (см. раздел 32).

3.2.2.10 **частичный контент элемента XML** (partial XML element content): Дочерние элементы XML определяются типом ACH.1 UNTAGGED, который обеспечивает часть контента элемента XML, сгенерированную включающим типом.

Примечание — Если включающий тип сам есть UNTAGGED, то этот включающий тип может быть также сгенерирован только частичным контентом элемента XML.

3.2.2.11 **команды кодирования с префиксами** (prefixed encoding instructions): Команды XER кодирования, которые назначаются с использованием префикса типа.

Примечание — Команды кодирования с префиксами могут удалять или заменять связанные команды кодирования типа, а также делать добавления к связанным командам кодирования типа.

3.2.2.12 **уточняющая информация** (qualifying information): Информация, предоставленная как часть спецификации цели для назначения команды кодирования, которая идентифицирует конкретные значения целевого типа.

3.2.2.13 **целевые команды кодирования** (targeted encoding instructions): Команды XER кодирования, которые назначаются с использованием целевого перечня в управляющем разделе XER кодирования.

Примечание — Целевые команды кодирования могут удалять или заменять связанные команды кодирования типа, а также делать добавления к связанным командам кодирования типа.

3.2.2.14 **атрибут идентификации типа** (type identification attribute): XML атрибут, который может появиться у любого элемента, чтобы идентифицировать тип этого элемента (см. раздел 36).

3.2.2.15 **унифицированный идентификатор ресурса (URI)** (Uniform Resource Identifier (URI)): Глобально однозначный идентификатор, назначаемый в соответствии с любой из URI схем, используемый для обеспечения идентификации пространства имен в EXTENDED-XER кодировании.

Примечание — URI схема, применяемая по умолчанию для ACH.1, дает возможность использовать значение объектного идентификатора ACH.1 для идентификации пространства имен (см. 15.9 и 28.1.5).

3.2.3 **действительный XML документ (для ACH.1 схемы)** (valid XML document (for an ACH.1 schema)): XML документ, который правильно построен (см. W3C XML 1.0) и чье содержание соответствует спецификации BASIC-XER, CXER или EXTENDED-XER для кодирования типа ACH.1, заданного ACH.1 схемой, возможно включая команды XER кодирования.

3.2.3.1 **команды XER кодирования** (XER encoding instructions): Команды кодирования, которые связаны с типом ACH.1 (или с компонентом типа ACH.1) путем назначения этого типа (или компонента) в префикс типа XER (см. ИСО/МЭК 8824-1, 30.3) или в управляющий раздел XER кодирования (см. ИСО/МЭК 8824-1, раздел 50).

3.2.3.2 **атрибут XML** (XML attribute): Часть кода EXTENDED-XER, состоящая из "XMLValue", заключенная в кавычки или апострофы, которой предшествует имя (атрибута) и знак равенства.

3.2.3.3 **элемент XML** (XML element): Часть XML документа, определенная в W3C XML 1.0.

Примечание — Элемент XML — это либо тег пустого элемента, либо начинается с начального тега и заканчивается конечным тегом. Начальный тег и тег пустого элемента могут содержать кодирования атрибутов.

3.2.3.4 **имя элемента XML** (XML element name): Лексическая единица, сопровождаемая "<" или "</" лексической единицей в связанных тегах.

3.2.4 **XML документ** (XML document): Последовательность символов, соответствующая определению документа в W3C XML 1.0.

3.2.5 **команда XML обработки** (XML processing instruction): Часть XML документа, которая содержит информацию относительно обработки части или всего этого документа (см. W3C XML 1.0).

Примечание — Команда обработки определяет тип обработки, для которого она применима и игнорируется другой обработкой. Это может быть использовано для идентификации таблицы стилей, которая должна быть применена, если документ предназначен для просмотра пользователем.

3.2.6 **XML пролог** (XML prolog): Начальная часть XML документа (которая не содержит информации о значении типа ACH.1, который закодирован).

4 Сокращения

В настоящем стандарте применяют следующие сокращения:

ASN.1 (ACH.1) — абстрактная синтаксическая нотация версии 1;
CXER — канонические правила XML кодирования;
PDU — единица обмена данными в соответствии с протоколом;
UCS — универсальный многооктетный набор кодированных символов;
URI — универсальный идентификатор ресурса;
UTC — универсальное глобальное время (по Гринвичу);
UTF-8 — UCS формат преобразования, 8-битовая форма;
XER — правила XML кодирования;
XML — расширяемый язык разметки.

5 Правила кодирования

5.1 Настоящий стандарт определяет три набора правил кодирования:
- базовые правила XML кодирования (BASIC-XER);
- канонические правила XML кодирования (CXER);
- расширенные правила XML кодирования (EXTENDED-XER).

5.2 Базовый набор правил кодирования BASIC-XER в общем случае не производит канонического кодирования и не обеспечивает любого пользователя управлением стилем XML, который он продуцирует.

5.3 Второй набор правил кодирования CXER производит каноническое кодирование. Данный набор правил кодирования определяется как ограничение возможности выбора, зависящего от реализации в BASIC-XER кодировании.

Примечание 1 — Любая реализация, соответствующая CXER для кодирования, совместима с BASIC-XER для кодирования. Любая реализация, соответствующая BASIC-XER для декодирования, совместима с CXER для декодирования. Следовательно, кодирование, произведенное в соответствии с CXER, есть кодирование, разрешенное BASIC-XER.

Примечание 2 — CXER производит кодирование при необходимости проверки достоверности абстрактного значения.

5.4 Третий набор правил кодирования, установленный в настоящем стандарте, — расширенные правила кодирования (EXTENDED-XER). Он определяется как вариации базовых правил кодирования BASIC-XER, установленные командами XER кодирования (см. 5.6), связанными с типом ACH.1. В отсутствие команд XER кодирования кодирование по расширенным правилам (EXTENDED-XER) отличается от кодирования по базовым правилам BASIC-XER только тем, что оно обеспечивает больше опций кодера.

5.5 Если тип, закодированный по CXER, содержит типы EMBEDDED PDV, EXTERNAL или CHARACTER STRING, внешнее кодирование перестает быть каноническим, если только кодирование для всех типов EMBEDDED PDV, EXTERNAL и CHARACTER STRING не было каноническим.

5.6 В настоящем стандарте установлены четыре команды кодирования

5.6.1 Настоящий стандарт устанавливает синтаксис и семантику команд XER кодирования (см. разделы 10—38). Команды XER кодирования влияют только на кодирование по расширенным правилам (EXTENDED-XER).

5.6.2 ACH.1 формирует базовую нотацию XML схемы. ACH.1 схема применяется, чтобы определить содержание и структуру данных, использующих ACH.1 и правила кодирования BASIC-XER (и CXER). Она может быть использована без команд XER кодирования.

5.6.3 Команды XER кодирования обеспечивают более широкую гибкость XML документов, чем она может быть изначально установлена в них.

5.6.4 Команды XER кодирования назначаются определениям типов ACH.1 или ссылкам на типы с использованием префиксов типа XER (см. ИСО/МЭК 8824-1, 30.3) или управляющего раздела XER кодирования (см. ИСО/МЭК 8824-1, 50), или и тому, и другому. Если команды кодирования связаны с определением типа, они передаются с типом ACH.1 (через ссылку на этот тип) в другие определения типа и другие модули ACH.1.Когда по расширенным правилам EXTENDED-XER кодируется тип, который имеет команды XER кодирования, связанные с некоторыми или всеми его частями, применяются эти окончательные команды кодирования и модифицируют результаты кодирования EXTENDED-XER.

Примечание — Окончательные команды кодирования также используются, когда выполняется подтверждение правильности и/или декодирование результатов кодирования EXTENDED-XER.

6 Соответствие

6.1 Динамическое соответствие для базовых правил XML кодирования специфицируется в разделе 7, динамическое соответствие для канонических правил XML кодирования специфицируется в разделе 8 и динамическое соответствие для расширенных правил XML кодирования специфицируется в разделе 9.

6.2 Статическое соответствие точно определено настоящим стандартом, предписывающим использование одного или более правил кодирования.

6.3 Альтернативное кодирование разрешается базовыми правилами XML кодирования и расширенными правилами XML кодирования как опция кодера. Декодеры, соответствующие BASIC-XER, должны поддерживать все альтернативы кодирования BASIC-XER. Декодеры, соответствующие EXTENDED-XER, должны поддерживать все альтернативы кодирования EXTENDED-XER.

Примечание — Этот раздел применяется независимо от наличия (или отсутствия) каких-либо окончательных команд кодирования.

6.4 Канонические правила XML кодирования не разрешают альтернативного кодирования для кодирования ACH.1 значений.

7 Базовые правила XML кодирования

7.1 Продукционные правила полного BASIC-XER кодирования

7.1.1 Соответствующий результат BASIC-XER кодирования — правильный XML документ, состоящий из:

а) XML пролога (который может быть пустым), как определено в 7.2;

б) элемента XML документа, который является законченным результатом кодирования значения единственного типа ACH.1, как определено в 7.3.

7.1.2 Спецификация в подразделах 7.2 — 7.6 полностью определяет BASIC-XER кодирование.

Примечание — Другие структурные компоненты, такие как команды XML обработки, не разрешены этими подразделами, и они никогда не создаются кодерами, соответствующими BASIC-XER.

7.1.3 XML документ должен быть кодирован с использованием UTF-8 для получения "строки октетов", которая формирует кодирование в соответствии с настоящим стандартом. Идентификатор объекта ACH.1 для этих правил кодирования определен в разделе 39.

7.1.4 Если в настоящем стандарте используется термин "пробел" ("white-space"), подразумевается один или несколько следующих символов стандарта Unicode: HORIZONTAL TABULATION (9), LINE FEED (10), CARRIAGE RETURN (13), SPACE (32). Цифры в круглых скобках — десятичные значения символов стандарта Unicode. Количество и выбор символов, которые составляют пробел — опция кодера.

7.1.5 Если настоящий стандарт использует термин "пробел с переходом" ("white-space with escapes"), подразумевается один или несколько символов, перечисленных в 7.1.4 с опцией кодера представляющей любые из этих символов с управляющей последовательностью в виде "&#n;" или "&#xn;" (см. ИСО/МЭК 8824-1, 11.15.8).

7.2 XML пролог

7.2.1 XML пролог должен:

а) быть пустым;

б) состоять из следующих последовательностей символов, расположенных по порядку, и, как опция кодера, последовательность может быть завершена разделителем (см. 7.1.4):

<?xml
version="1.0"
encoding="UTF-8"?>

7.2.2 Последовательности символов в 7.2.1 не должны содержать разделитель, но должны быть разделены одним символом SPACE (32).

7.3 Элемент XML документа

7.3.1 Элементом XML документа будет "XMLTypedValue", как определено в ИСО/МЭК 8824-1, пункт 15.2, с изменениями и ограничениями, установленными в последующих подразделах раздела 7.3.

7.3.1.1 Все появления "ExternalTypeReference" в "XMLTypedValue" должны быть заменены "typereference" в этом "ExternalTypeReference".

7.3.2 Лексический элемент ACH.1 "comment" (см. ИСО/МЭК 8824-1, пункт 11.6) не должен присутствовать. Если XER кодирование содержит пару смежных дефисов "/*" или "*/", это должно интерпретироваться как часть данных, а не как разделители комментариев ACH.1.

7.3.3 Если ИСО/МЭК 8824-1, 11.1.4, 11.11 и 11.13 разрешают использование пробелов ACH.1 между лексическими единицами в "xmlbstring" или в "xmlhstring", использованные символы должны быть ограничены до "разделителей", определенных в 7.1.4.

7.3.3.1 "XMLBooleanValue", определенный в ИСО/МЭК 8824-1 17.3, должен быть только "EmptyElementBoolean", а "XMLSequenceOfValue" и "XMLSetOfValue" с компонентом булевского типа должен быть "ValueList".

7.3.4 "XMLIntegerValue", определенный в ИСО/МЭК 8824-1, 18.9, должен быть только "XMLSignedNumber".

7.3.4.1 "XMLEnumeratedValue", определенный в ИСО/МЭК 8824-1, 19.8, должен быть только "EmptyElementEnumerated", а "XMLSequenceOfValue" с "XMLSetOfValue" с компонентом перечислимого типа должен быть "ValueList".

7.3.4.2 "XMLSpecialRealValue", определенный в ИСО/МЭК 8824-1, 20.6, должен быть только "EmptyElementReal".

7.3.5 Значение "XMLBitStringValue", определенное в ИСО/МЭК 8824-1, 21.9, не должно быть "XMLIdentifierList".

7.3.6 Значение "XMLExternalValue", определенное ИСО/МЭК 8824-1, 34, должно быть заменено на "XMLExternalValue" по 8.4.

7.4 Кодирование типа EXTERNAL

7.4.1 Получение значения "XMLExternalValue", используемого для XER кодирования внешнего типа, должно быть значением "XMLValue" для кодирования типов "последовательности", определенных в ИСО/МЭК 8825-2, 26.1, со значением, определенным в 26.2—26.4 указанного стандарта.

Примечание — XER кодирование типа "EXTERNAL" неэквивалентно XML нотации значений, определенной в ИСО/МЭК 8824-1.

7.4.2 Допускается применять ИСО/МЭК 8825-2, 26.5 – 26.8, но пункт 26.6 должен быть заменен на 7.4.3 настоящего стандарта.

7.4.3 Если значение данных — значение типа "single ASN.1 type" и кодирующие правила для значения этих данных определены в настоящем стандарте, то реализация сообщений должна использовать выбранный тип "single ASN.1 type".

7.4.4 Допускается применять ИСО/МЭК 8825-2, 26.9 — 26.11, но пункт 26.6 должен быть заменен на 7.4.5 настоящего стандарта. Примечание по 26.9 ИСО/МЭК 8825-2 применимо.

7.4.5 Если выбирают тип кодирования "single-ASN.1-type", то тип ACH.1 должен быть "XMLTypedValue" типа кодирования "EXTERNAL" со значением, эквивалентным значению данных, которые будут кодированы.

7.5 Кодирование открытого типа

Могут быть использованы обе альтернативы "XMLOpenTypeFieldVal" (см. ИСО/МЭК 8824-2, пункт 14.6)

Примечание — Использование альтернативы "xmlhstring" в общем случае не рекомендуется, так как не существует механизмов идентификации правил кодирования, используемых для создания "xmlhstring" в отдельном случае кодирования. Примерами, где такая альтернатива может быть удобной — случаи, когда сообщение, кодируемое в XER (например, для целей вывода на дисплей), является результатом предыдущего двоичного кодирования и полностью не декодировано или когда имеется двустороннее соглашение.

7.6 Декодирование типов с маркерами расширения

7.6.1 Декодер BASIC-XER будет воспринимать как правильный XML документ результат BASIC-XER кодирования типов с маркерами расширения, в которых имеются неизвестные расширения.

7.6.2 Неизвестные расширения в последовательности или типе набора приводят к неожидаемым элементам XML с именами, отличными от имен следующего ожидаемого элемента XML.

7.6.3 Неизвестные расширения в типе выбора приводят к единственному неожидаемому элементу XML вместо элемента, соответствующего одному из известных выборов. Он всегда будет иметь имя элемента XML , отличное от имени любого элемента XML, который кодирует известную альтернативу типа выбора.

7.6.4 Неизвестные расширения в перечислимом типе приводят к элементу XML с неожидаемым содержанием, но без неожидаемых элементов XML.

7.6.5 Неизвестные расширения, возникающие при смягчении ограничений на подтип, приводят к кодированию, которое может быть правильным кодированием любого типа, не имеющего ограничений. Такое кодирование может продуцировать неожидаемое содержание, но без неожидаемых элементов XML.

8 Канонические правила XML кодирования

В случае если "XMLTypedValue" содержит возможности выбора, настоящий раздел допускает определение строго одного варианта выбора для получения уникального кодирования. Условия настоящего раздела определяют канонические правила XML кодирования.

8.1 Общие правила для канонического XER

8.1.1 XML пролог должен быть пустым (см. 7.2.1).

8.1.2 Все лексические единицы, формирующие "XMLTypedValue", не должны иметь пробел ("white-space") между ними (см.7.3.3).

Примечание — Это правило гарантирует, что цифровая подпись документа может быть легко сгенерирована без учета любой возможной вставки разделителя между лексическими элементами "XMLTypedValue".

8.1.3 ESC-последовательности, описанные в ИСО/МЭК 8824-1, 11.15.8, не используют.

8.1.4 Если нотация XML значений разрешает использовать пустой XML тег (см. ИСО/МЭК 8824-1, 15.5, 16.8), то должен использоваться этот пустой тег.

8.2 Действительные значения

8.2.1 Действительное значение нуля должно быть кодировано как "0".

8.2.2 Для всех других действительных значений ограничения, применяемые к "realnumber" (см. ИСО/МЭК 8824-1, 11.9), определяют следующие подпункты.

8.2.3 Значение "Realnumber" должно быть нормализовано так, чтобы его целая часть состояла из одной цифры, отличной от нуля. Десятичная точка должна присутствовать и должна сопровождаться дробной частью, содержащей как минимум одну цифру (может быть нулем). Дробная часть не должна содержать замыкающие нули, следующие после первой цифры.

8.2.4 Дробная часть значения "Realnumber"должна сопровождаться "E" (не "e") и показателем степени (может быть нулем).

Примечание — Использование ведущих нулей в показателе степени запрещено ИСО/МЭК 8824-1, 11.9.

8.2.5 Знак "+" не должен присутствовать ни перед целой частью значения "Realnumber", ни перед показателем степени.

8.3 Значение "битовая строка"

8.3.1 Если альтернатива "XMLTypedValue" для "XMLBitStringValue" (см. ИСО/МЭК 8824-1, 21.9) может быть использована (как определено в ИСО/МЭК 8824-1, 21.10), то она должна быть использована. В противном случае должна быть использована альтернатива "xmlbstring" со всеми удаленными пробелами "white-space" (см. 7.3.3).

8.3.2 Если тип битовой строки — "NamedBitList", то не должно быть битов с замыкающими нулями (см. ИСО/МЭК 8824-1, 21.7).

8.4 Значение строки октетов (octetstring)

Если альтернатива "XMLTypedValue" для "XMLOctetStringValue" (см. ИСО/МЭК 8824-1, 22.3) может быть использована (как определено в ИСО/МЭК 8824-1, 22.4), то она должна быть использована. В противном случае должна быть использована альтернатива "xmlbstring" со всеми удаленными пробелами "white-space" (см. 7.3.3) и все буквы должны быть заглавными.

8.5 Значение "последовательность"

Все компоненты последовательности, имеющие значение "по умолчанию" и набор абстрактных значений к значениям "по умолчанию" должны иметь кодированные значения "по умолчанию", представленные в виде текста. Эти компоненты всегда кодируются.

8.6 Значение "множество"

8.6.1 Тип набора будет иметь элементы в его "RootComponentTypeList", отсортированном в каноническом порядке, определенном в ИСО/МЭК 8824-1, 8.6, и дополнительно с тегом ACH.1 для целей определения порядка, в котором компоненты закодированы, когда один или более компонентов является типом выбора. Каждый такой тип выбора расположен по порядку так, как если бы он имел тег, равный наименьшему тегу в "RootAlternativeTypeList" того типа выбора или любого такого типа выбора, который встроен в него.

8.6.2 Элементы типа "множество", встречающиеся в "RootComponentTypeList", кодируют в порядке окончательной сортировки. После кодирования элементов из "RootComponentTypeList" (если имеются) элементы множества, входящие в "ExtensionAdditionList", должны быть кодированы в порядке, в котором они определены (пример такого упорядочения элементов представлен в ИСО/МЭК 8825-2, 20).

8.6.3 Все компоненты множества, имеющие значение "по умолчанию" и набор абстрактных значений к значениям "по умолчанию", должны иметь кодированные значения "по умолчанию", представленные в виде текста. Эти компоненты всегда кодируются.

8.7 Значение "множество-из"

8.7.1 Порядок элементов "XMLSetOfValue" (см. ИСО/МЭК 8824-1, 27.3) будет определяться сортировкой строк символов, которую обеспечивает CXER кодирование для каждого элемента, как определено в 8.7.2 и 8.7.3.

8.7.2 Порядок сортировки для символьных строк определен использованием 32-битного значения символов (по ИСО/МЭК 106461), причем символы с меньшим кодовым значением предшествуют символам с большим кодовым значением.

8.7.3 Условный символ-заполнитель используют для определения порядка сортировки. Этот символ предшествует всем другим символам. При определении, какую из строк, "A" или "B", ставить ранее, ставят более короткую строку, дополняя ее условным символом-заполнителем в конце строки (если необходимо). Строку "A" ставят ранее строки "B", только если символ в строке "A" превосходит (см. 8.7.2) соответствующий символ в строке "B", причем сравнивают значения первых символов, с которых эти строки начинают различаться.

8.8 Значение "идентификатор объекта"

Значение "XMLObjIdComponent" (см. ИСО/МЭК 8824-1, 31.3) должно быть "XMLNumberForm".

8.9 Значение "относительный идентификатор объекта"

Значение "XMLRelativeOIDComponent" (см. ИСО/МЭК 8824-1, 32.3) должно быть "XMLNumberForm".

8.10 Обобщенное время

8.10.1 Кодирование значения типа "GeneralizedTime" должно заканчиваться символом "Z" (см. ИСО/МЭК 8824-1, 42.3).

8.10.2 Строка, отображающая секунды, должна присутствовать всегда.

8.10.3 Если присутствует строка, отображающая доли секунды, то в ней пропускают завершающие нули. Если дробная часть соответствует “0”, то ее пропускают вместе с десятичной точкой.

Например, секунды, представленные строкой "26.000", кодируют как "26". Секунды, представленные строкой "26.5200", кодируют как "26.52".

8.10.4 Если имеется десятичная точка, она должна быть представлена в виде “.”.

8.10.5 Полночь (GMT) кодируют как строку вида:

“YYYYMMDD000000Z”,

где “YYYYMMDD” отображают день, следующий за полночью.

Например, следующие кодирования являются правильными:

"19920521000000Z"

"19920622123421Z"

"19920722132100.3Z"

Следующие кодирования являются неправильными:

"19920520240000Z" (полночь представлена некорректно)

"19920622123421.0Z" (ложные замыкающие нули)

"19920722132100.30Z" (ложные замыкающие нули)

8.11 Универсальное синхронизированное время

8.11.1 Кодирование типа "UTCTime" должно заканчиваться символом "Z" (см. ИСО/МЭК 8824-1, 43.3).

8.11.2 Строка, отображающая секунды, должна присутствовать всегда.

8.11.3 Полночь (GMT) кодируют как строку вида:

"YYMMDD000000Z",

где "YYMMDD" представляют день, следующий за полночью.

Например, следующие кодирования являются правильными:

"920521000000Z"

"920622123421Z"

"920722132100Z"

Следующие кодирования являются неправильными:

"920520240000Z" (полночь представлена некорректно)

"9207221321Z" (секунды "00" пропущены)

8.12 Значение открытого типа

Альтернатива "xmlhstring" для "XMLOpenTypeFieldVal" не будет использоваться (см. 7.5).

9 Правила расширенного XML кодирования

9.1 Общие положения

9.1.1 Расширенные правила XML кодирования (EXTENDED-XER) дополняют и модифицируют базовые правила BASIC-XER. Они предоставляют возможность ACH.1 определять форму и содержание значительно более широкого диапазона XML документов.

9.1.2 EXTENDED-XER расширяют возможности BASIC-XER в трех основных направлениях:

а) обеспечение дополнительных опций кодера (например, для вставки команд обработки XML или комментариев XML и для использования идентификаторов для наименования битов в значении строки битов);

б) определение набора команд кодирования, которые могут быть использованы для определения модификации BASIC-XER кодирования типа ACH.1, включая команду кодирования для использования простого текста вместо тегов пустых элементов для булевских, целых (с поименованными числами), нумерованных, специальных значений вещественных чисел типов и типов строк битов (с именованными битами).

в) проигнорирование неизвестных атрибутов из управляющего пространства имен (например "schemaLocation") при отсутствии команд кодирования, для чего необходимы декодеры или некоторые известные атрибуты, которые могут быть вставлены другими инструментами XML, причем значения последних могут отличаться от значений, вставленных соответствующими кодерами, например атрибут идентификации типа (см. 9.2.10).

9.1.3 Если спецификация ACH.1 не содержит никаких команд XER кодирования, то каждое BASIC-XER кодирование любого абстрактного значения типа ACH.1 будет также EXTENDED-XER кодированием того же самого абстрактного значения этого типа.

Примечание — Обратное — не справедливо. Даже при отсутствии команд XER кодирования EXTENDED-XER кодирование не всегда будет соответствовать BASIC-XER кодированию (см. 9.1.2 а и 9.1.2 в).

9.1.4 Все появления нотации "Type" ACH.1 имеют связанный набор (возможно, пустой) команд XER кодирования (окончательные связанные команды кодирования). Команды кодирования ассоциированы с "Type" через:

а) унаследованные команды кодирования — присутствие связанных команд кодирования в "Type", применяемом в определении "typereference", используемом как "Type";

б) целевые команды кодирования — назначение одной или нескольких команд XER кодирования появлению "Type" с использованием управляющего раздела XER кодирования (см. ИСО/МЭК 8824-1, 50);

Примечание — Модуль ACH.1 может содержать только один управляющий раздел XER кодирования и, следовательно, только один перечень "EncodingInstructionAssignmentList" (см. ИСО/МЭК 8824-1, 50.2)

в) команды кодирования с префиксом — назначение одной или нескольких команд XER кодирования появлению "Type" с использованием префиксов типа XER (см. ИСО/МЭК 8824-1, 50);

г) команды кодирования из перечня импортируемых — назначение одной или нескольких команд XER кодирования ссылкам всех типов, импортируемых из идентифицированного ACH.1 модуля.

9.1.5 Действием назначения команды XER кодирования является добавление, удаление или замещение связанных команд кодирования (см. раздел 14 для правил, применяемых для множественного назначения команд XER кодирования).

9.1.6 Порядок (или метод), в котором команды кодирования становятся частью (или удаляются из) набора связанных команд кодирования, не является существенным в применении окончательных команд кодирования.

9.1.7 Окончательные команды кодирования влияют на EXTENDED-XER кодирование типов. Никакого другого влияния они не оказывают и, в частности, не являются связанными ни с какими либо ссылками на значения, определенными при использовании типа, не влияют на отображения значений и на другие правила кодирования.

Примечание — Имеются, однако, требования к однозначности имен, на которую влияют присутствие окончательного имени (NAME), пространства имен (NAMESPACE) или нетегированной (UNTAGGED) команды кодирования. Эти требования могут интерпретироваться либо как ограничения способа, которым типы с такими окончательными командами кодирования могут использоваться, либо как ограничения использования этих команд кодирования.

9.2 Соответствие EXTENDED-XER

9.2.1 Если по спецификации ACH.1 инструкции XER кодирования назначаются в соответствии с разделами 10—16, так что тип или компонент ACH.1 имеют окончательные команды кодирования, которые нарушают ограничения, определенные в разделах 17 и последующих, то эта спецификация ACH.1 не согласуется с данным Международным стандартом, даже если (без команд XER кодирования) она соответствует всем требованиям ИСО/МЭК 8824-1.

Примечание — Назначение команды кодирования "Type" может быть только случайно неуточненным, так как она может быть сделана недействительной (удаленной из набора связанных команд кодирования) последующим назначением. Обычно окончательная команда кодирования определяет соответствие спецификации. В некоторых случаях окончательная команда кодирования, не применимая к типу, к которому ее пытаются применить, игнорируется. Если разделы, определяющие синтаксис и применение команд кодирования, идентифицируют обстоятельства, при которых команда кодирования игнорируется в применении окончательных команд кодирования, то разделы, специфицирующие кодирование, обычно не упоминают возможное наличие этой окончательной команды кодирования.

9.2.2 Соответствующее EXTENDED-XER кодирование типа ACH.1 без окончательных команд кодирования будет кодированием, произведенным по Базовым правилам XML кодирования (BASIC-XER), специфицированным в разделе 8 с дополнительными опциями кодера, определенными в 9.2.5 и 9.2.6.

Примечание — Чтобы принять и обработать декларацию типа документа W3C XML, требуется наличие декодеров EXTENDED-XER (см. 9.2.4), но они не генерируются соответствующими кодерами и не составляют части EXTENDED-XER кодирования.

9.2.3 EXTENDED-XER кодирование типа ACH.1 с окончательными командами кодирования или с компонентами (на любой глубине и после разрешения всех ссылок на типы), которые имеют связанные команды кодирования, будет кодированием, специфицированным в разделе 16.

Примечание — Окончательные команды кодирования применяются в EXTENDED-XER кодировании и также используются декодерами и средствами проверки достоверности EXTENDED-XER кодирования.

9.2.4 Декодеры EXTENDED-XER (при использовании команды MODIFIED-ENCODINGS — см. 25) будут обрабатывать декларацию любого типа документа (см. W3C XML 1.0, 2.8), которая присутствует в соответствии с требованиями XML процессора, не осуществляющего подтверждение правильности (см. W3C XML 1.0, 5.1). Такая обработка будет выполняться (концептуально) до применения всех других требований декодирования в этом Международном стандарте. Кодеры EXTENDED-XER не будут включать декларацию типа документа.

9.2.5 Кодер EXTENDED-XER может (в качестве опции кодера) вставлять команды обработки XML или комментарии XML (в дополнение к любым, которые могут потребоваться в соответствии с разделом 29) в элемент документа XML или пролог XML в любую позицию, разрешенную W3C XML 1.0. Синтаксическая форма и семантика команд обработки XML определены в W3C XML 1.0, 2.6. Синтаксическая форма и семантика комментариев XML определены в W3C XML 1.0, 2.5.

9.2.6 Если команда кодирования GLOBAL-DEFAULTS с ключевым словом MODIFIED-ENCODINGS отсутствует (см. 25) в управляющем разделе XER кодирования, то:

а) "XMLIntegerValue" (см. ИСО/МЭК 8824-1, 18.9) может быть либо "XMLSignedNumber", либо "EmptyElementInteger" как опция кодера;

б) "XMLBitStringValue" (см. ИСО/МЭК 8824-1, 21.9) может быть любой альтернативой этого продукционного правила как опция кодера. Если используется "XMLIdentifierList", то это "EmptyElementList".

9.2.7 Если в управляющем разделе XER кодирования имеется команда кодирования GLOBAL-DEFAULTS с ключевым словом MODIFIED-ENCODINGS (см. раздел 25), то:

а) "XMLBooleanValue" (см. ИСО/МЭК 8824-1, 17.3) должно быть "TextBoolean";

б) "ExtendedXMLIntegerValue" (см. 16.4) должно быть альтернативой "ModifiedXMLIntegerValue", определенной в 16.8;

Примечание — Это разрешает использование текстового значения для "NamedNumber" целого типа, как опцию кодера, но также модифицирует синтаксис для численного кодирования целого значения.

в) "ExtendedXMLEnumeratedValue" (см. 33.3) не должно быть "EmptyElementEnumerated";

Примечание — В отсутствие команды GLOBAL-DEFAULTS ключевого слова MODIFIED-ENCODINGS "ExtendedXMLEnumeratedValue" не может быть "TextEnumerated" (см. 7.3.4.1 и 33.3).

г) "ExtendedXMLRealValue" (см. 16.4) должно быть альтернативой "ModifiedXMLRealValue", определенной в 16.9;

д) "XMLSpecialRealValue" (см. ИСО/МЭК 8824-1, 20.6) должно быть альтернативой "TextReal";

е) альтернативой "XMLIdentifierList" в "XMLBitStringValue" (см. ИСО/МЭК 8824-1, 21.9) должно быть "TextList" (см. 9.2.8 б);

ж) "XMLSequenceOfValue" и "XMLSetOfValue" (см. ИСО/МЭК 8824-1, 24.3 и 26.3) должно быть "XMLDelimitedItem" для всех типов компонентов, причем Таблица 5 не учитывается (см. ИСО/МЭК 8824-1, 25.5);

з) "xmlhstring" (см. ИСО/МЭК 8824-1, 11.13) не должно содержать пробелов (см. 7.1.4);

и) все появления пробелов, которые либо вне XML тегов, либо внутри значений XML атрибутов, могут быть пробелом с переходом (см. 7.1.5) как опция кодера.

Примечание — Существует несколько команд кодирования (таких как UNTAGGED), которые не могут быть использованы, за исключением команды GLOBAL-DEFAULTS ключевого слова MODIFIED-ENCODINGS.

9.2.8 Если команда кодирования GLOBAL-DEFAULTS с ключевым словом MODIFIED-ENCODINGS (см. 25) присутствует в управляющем разделе XER кодирования, то EXTENDED-XER кодер может (в качестве опций кодера):

а) использовать альтернативу "TextInteger" для "ModifiedXMLIntegerValue" (см. 16.8), при условии, что существует "NamedNumber" для целого значения в определении типа (см. также 9.2.7 б);

Примечание — Использование этого вида кодирования с поименованными значениями, который был добавлен в более позднюю версию, может сделать абстрактные значения нечитаемыми при внедрении более ранней версии спецификации.

б) использовать "XMLIdentifierList" (см. ИСО/МЭК 8824-1, 21.9) для "XMLBitStringValue", при условии, что значение битовой строки, которая должна быть закодирована, не содержит непоименованных битов (см. также 9.2.7 е);

Примечание — Использование этого вида кодирования с поименованными битами, который был добавлен в более позднюю версию, может сделать абстрактное значение невидимым более ранней версией спецификации.

9.2.9 Разрешенные в EXTENDED-XER кодировании опции кодера будут полностью восприняты соответствующими декодерами и средствами проверки достоверности.

9.2.10 Соответствующие декодеры и средства проверки достоверности должны воспринимать, но могут и проигнорировать наличие атрибута из управляющего пространства имен в любом XML элементе кодирования, если его наличие и использование не определены в разделах 36 и 37. Кодеры не должны генерировать такие атрибуты, если это не определено в разделах 36 и 37.

Примечание — Другие XML инструменты могут вставлять такие атрибуты. В общем случае декодер EXTENDED-XER не может легко определить разрешенное значение и смысл некоторых атрибутов из управляющего пространства имен. Их наличие и значение могут быть использованы в некотором приложении, если, например, присутствуют непредвиденные дочерние элементы XML, которые скорее передаются приложению (как опция декодера), чем игнорируются или продуцируют фатальную ошибку декодера.

9.2.11 Спецификация ACH.1 неправомерна, если для декодера нет возможности определить однозначно для абстрактных значений (используя только имя тега XML и контент любого предыдущего элемента XML) компонент ACH.1 (или маркер расширения), с которым связан элемент XML.

Примечание 1 — Ассоциация не может зависеть от контента элемента XML, или его атрибутов, или от любого последующего элемента XML.

Примечание 2 — Это условие всегда удовлетворяется, когда нет команд XER кодирования, но неправильное применение UNTAGGED для уничтожения связанных тегов вокруг, например повторений, (последовательность-из или набор-из) и альтернатив (выбор) и неправильное использование NAME может привести к неправомочным спецификациям.

Примечание 3 — В 9.2.11 приведено необходимое условие правильного кодирования, но принято считать, что, в общем случае, невозможно для инструментов ACH.1 (или автора) проверить правомочность на основании этого единственного оператора высокого уровня. В Приложении В приведена модель эффекта применения UNTAGGED и правила, которые могут обеспечить правомочность спецификации, определенной в 9.2.11.

9.2.12 Если спецификация ACH.1 содержит типы "ObjectClassFieldType", являющиеся открытыми типами (см. ИСО/МЭК 8824-2, 14.2), с табличными ограничениями или ограничениями типов, такие ограничения будут игнорироваться при использовании условия, приведенного в 9.2.11.

9.3 Структура результата расширенного XML кодирования

9.3.1 Законченное расширенное XML кодирование продуцирует хорошо сформированный XML документ, состоящий из:

а) XML пролога (может быть пустым как опция кодера), как определено в 7.2;

б) элемента XML документа, являющегося результатом законченного кодирования значения единственного типа ACH.1, называемого корневым типом, как определено в разделе 16.

9.3.2 Результаты кодирования "XML Value", используемые для базового XML кодированиия, модифицированы для расширенного XML кодирования путем применения окончательных команд кодирования для типов ("Type"), которые они кодируют, и окончательных команд кодирования для их компонентов (до любой глубины) вместе с любыми командами кодирования GLOBAL-DEFAULTS.

Примечание — В исключительных случаях полное содержание элемента XML документа для сильно уплотненной структуры ACH.1 может (путем использования команды кодирования UNTAGGED) состоять не более чем из линейной последовательности XML элементов, в которой только корневой элемент может иметь дочерние элементы. Применение команды UNTAGGED ограничено, чтобы обеспечить однозначное отображение таких результирующих линейных последовательностей XML элементов в компоненты абстрактного значения корневого типа ACH.1 (см. 9.2.11).

9.3.3 Элемент XML документа при расширенном XER кодировании состоит из единственного XML элемента, который должен быть "ExtendedXMLTypedValue" для кодируемого типа (корневой тип). Он может включать атрибуты в стартовом теге или в теге пустого элемента и может иметь контент, который включает и дочерние элементы (см. W3C XML), и нетегированный текст. Дочерние элементы сами могут иметь свои дочерние элементы и нетегированный текст.

9.3.4 Абстрактные значения компонентов включающего типа кодируются как значения "ExtendedXMLValue" (см. 16.4), возможно модифицированные командами кодирования, примененными к ним или к их собственным компонентам. Эти значения "ExtendedXMLValue":

а) могут быть охвачены тегами XML стартовым и конечным (называемыми ассоциативными тегами), чтобы сформировать элемент в пределах "ExtendedXMLValue" включающего типа;

б) могут (путем использования команды кодирования UNTAGGED для типа, который не является посимвольно-кодируемым) формировать частичный контент XML элемента для значения "ExtendedXMLValue" включающего типа;

Примечание — Приложение В описывает результат применения команды UNTAGGED как продукционных правил для контента частичного XML элемента, который может сочетаться с другими результатами кодирования для того, чтобы сформировать контент XML элемента, чей тип не является UNTAGGED.

в) могут (путем использования команды кодирования UNTAGGED для посимвольно-кодируемого типа) формировать полное значение "ExtendedXMLValue" компонента;

г) могут (путем использования команды кодирования ATTRIBUTE для посимвольно-кодируемого типа) формировать значение "CharacterEncodableValue" в "QuotedValue" атрибута "Attribute" (см. 19.3.3).

9.3.5 Если значение "ExtendedXMLValue" — пусто и его ассоциированные теги не удалены путем использования команды кодирования UNTAGGED, то ассоциированные предшествующий и последующий теги (как опция кодера) могут быть заменены пустым от элементов тегом XML (см. ИСО/МЭК 8824-1, 16.8). Он называется ассоциированным пустым от элементов тегом.

9.3.6 Преобразование, определенное в 9.3.5, выполняется концептуально после завершения полного процесса кодирования и может быть предотвращено командой кодирования PI-OR-COMMENT

(см. раздел 29), продуцирующей одну или более команд обработки XML или один или несколько элементов комментария XML между стартовым и конечным тегами.

9.3.7 Ассоциированный предшествующий тег, ассоциированный последующий тег и ассоциированный пустой от элементов тег в совокупности называются ассоциированными тегами. Имена XML элементов в ассоциированных тегах называются именами ассоциированных тегов и (в отсутствие окончательных команд кодирования NAME и NAMESPACE) являются идентификаторами, референтными именами типов или именами типов "xmlasn1typename" (см. ИСО/МЭК 8824-1, 11.25).

10 Нотация, набор символов и лексические единицы, использованные в командах XML кодирования

10.1 Нотация, используемая при спецификации синтаксиса команды кодирования "EncodingInstruction" в префиксе типа XER (см. раздел 12) и в перечне "EncodingInstructionAssignmentList" в управляющем разделе XER кодирования (см. раздел 13), определена ИСО/МЭК 8824-1, 5.

10.2 ИСО/МЭК 8824-1, 10 применяется к "EncodingInstruction" XER и к "EncodingInstruction-AssignmentList" XER.

Примечание — В частности, символы пробела АСН.1 могут произвольно появиться между лексическими единицами в обеих этих синтаксических конструкциях, если только не используется нотация "&" (см. ИСО/МЭК 8824-1, 5.4).

10.3 Общие правила, определенные в ИСО/МЭК 8824-1, 11.1, также применяются к "EncodingInstruction" XER и к "EncodingInstructionAssignmentList" XER.

Примечание — В частности, комментарий АСН.1 может быть использован везде, где разрешены пробелы АСН.1, и требования для вставки пробелов и комментариев, которые в других обстоятельствах могли бы быть неправильно восприняты, специфицированы в ИСО/МЭК 8824-1.

10.4 Следующие лексические единицы используются в настоящем стандарте:

comment (см. ИСО/МЭК 8824-1, 11.6);
cstring (см. ИСО/МЭК 8824-1, 11.14);
identifier (см. ИСО/МЭК 8824-1, 11.3);
modulereference (см. ИСО/МЭК 8824-1, 11.5);
number (см. ИСО/МЭК 8824-1, 11.8);
typereference (см. ИСО/МЭК 8824-1, 11.2);
"{" (см. ИСО/МЭК 8824-1, 11.26);
"}" (см. ИСО/МЭК 8824-1, 11.26);
"." (см. ИСО/МЭК 8824-1, 11.26);
":" (см. ИСО/МЭК 8824-1, 11.26);
"," (см. ИСО/МЭК 8824-1, 11.26);
";" (см. ИСО/МЭК 8824-1, 11.26);
""" (см. ИСО/МЭК 8824-1, 11.26);
"*" (см. ИСО/МЭК 8824-1, 11.26);
"::=" (см. ИСО/МЭК 8824-1, 11.16);
"<" (см. ИСО/МЭК 8824-1, 11.26);
">" (см. ИСО/МЭК 8824-1, 11.26);
"</" (см. ИСО/МЭК 8824-1, 11.21);
"/>" (см. ИСО/МЭК 8824-1, 11.22).

Дополнительные лексические единицы ("ModifiedXMLNumber" и "ModifiedXMLRealNumber") определены и использованы в 16.8.3 и 16.9.

11 Ключевые слова

11.1 Слова, специфицированные ниже в 11.3 и 11.4, используются в одной из двух или в каждой конструкции "EncodingInstruction" и XER "EncodingInstructionAssignmentList" (в дополнение к некоторым зарезервированным словам АСН.1) и могут появляться в этих синтаксических конструкциях только со значением, присвоенным им в последующих разделах этого Международного стандарта. Исключения определены в 11.2.

11.2 Ключевые слова не являются зарезервированными, но если "typereference" АСН.1, который совпадает с ключевым словом, приведенным в 11.3, требуется в XER "EncodingInstruction" или XER "EncodingInstructionAssignmentList", то должны быть применены продукционные правила "ModuleAndTypeReference".

11.3 Ключевыми словами являются:

AFTER-TAG
AFTER-VALUE
ANY-ATTRIBUTES
ANY-ELEMENT
AS
ATTRIBUTE
BASE64
BEFORE-TAG
BEFORE-VALUE
CAPITALIZED
COLLAPSE
CONTROL-NAMESPACE
DECIMAL
DEFAULT-FOR-EMPTY
ELEMENT
EMBED-VALUES
GLOBAL-DEFAULTS
IN
LIST
LOWERCASED
MODIFIED-ENCODINGS
NAME
NAMESPACE
NOT
PI-OR-COMMENT
PREFIX
REPLACE
TEXT
UNCAPITALIZED
UNTAGGED
UPPERCASED
USE-NIL
USE-NUMBER
USE-ORDER
USE-QNAME
USE-TYPE
USE-UNION
WHITESPACE

11.4 Дополнительные ключевые слова используются в продукционном правиле "BuiltInTypeName" (см. 13.2.3), но все они являются зарезервированными словами АСН.1 (см. ИСО/МЭК 8824-1, 11.27) и никогда не могут быть использованы в АСН.1 как "typereference".

12 Назначение команды XML кодирования типу АСН.1 с использованием префикса типа

12.1 Окончательные команды кодирования для типа могут:

а) требовать использование альтернатив "ExtendedXMLValue", которые не являются альтернативами "XMLValue" для этого типа;

Примечание — Альтернативы продукционного правила "ExtendedXMLValue" включают и (неизмененные) альтернативы продукционного правила "XMLValue", используемого в BASIC-XER, и альтернативные продукционные правила, выбранные командами XER кодирования.

б) изменить имя ассоциативного тега, "AttributeName" или значение атрибута идентификатора типа для кодирования этого типа;

в) послужить причиной того, что "ExtendedXMLValue" компонента типа ACH.1 будет вставлено как "CharacterEncodableValue" в "QuotedValue" атрибута ("Attribute") (см. 19.3.3);

г) определить имя пространства имен XML для референтных имен типа и идентификаторов, определенных в модуле ACH.1, и рекомендовать префикс пространства имен для использования с этим пространством имен;

д) определить, когда имя, ограниченное пространством имен (вместо неограниченного имени) должно быть использовано в XML элементе или как имя XML атрибута;

е) определить удаление связанных тегов, имеющее обычно результатом либо нетегированный текст, либо контент частичного XML элемента (которому может предшествовать или за которым может следовать другой контент частичного XML элемента — см. приложение В);

ж) определить вставку одной или более команд обработки XML или одного или более комментария XML (см. раздел 29):

- перед связанным предшествующим тегом или связанным пустым от элемента тегом;
- между связанным предшествующим тегом и "ExtendedXMLValue";

Примечание 1 — Это препятствует использованию ассоциированного пустого от элемента тега.

- между "ExtendedXMLValue" и связанным последующим тегом;

Примечание 2 — Это препятствует использованию ассоциированного пустого от элемента тега.

- после ассоциированного последующего тега.

Примечание 3 — Все четыре варианта вставки препятствуют использованию UNTAGGED при удалении ассоциированных тегов (см. 29.2.2).

12.2 Команды XER кодирования могут быть назначены типам ACH.1 с использованием либо продукционного правила "EncodingInstruction" в префиксе типа XER, либо продукционного правила "EncodingInstructionAssignmentList" в управляющем разделе кодирования XER. Назначение с использованием префикса типа определено в этом разделе. Назначение с использованием управляющего раздела копирования XER описано в разделе 13.

Примечание — Действие множественных назначений команд кодирования той же самой категории описано в разделе 14.

12.3 Продукционным правилом команды кодирования XER "EncodingInstruction" является:

```
EncodingInstruction::=
    PositiveInstruction
    | NegatingInstruction
PositiveInstruction::=
    AnyAttributeInstruction
    | AnyElementInstruction
    | AttributeInstruction
    | Base64Instruction
    | DecimalInstruction
    | DefaultForEmptyInstruction
    | EmbedValuesInstruction
    | GlobalDefaultsInstruction
    | ListInstruction
    | NameInstruction
    | NamespaceInstruction
    | PIOrCommentInstruction
    | TextInstruction
    | UntaggedInstruction
    | UseNilInstruction
    | UseNumberInstruction
    | UseOrderInstruction
    | UseQNameInstruction
    | UseTypeInstruction
```

| UseUnionInstruction
| WhitespaceInstruction
NegatingInstruction::=
NOT PositiveInstruction
| ElementInstruction

12.4 "ElementInstruction" (см. раздел 19) является строгим синонимом для NOT UNTAGGED и не обсуждается в этом разделе.

Примечание 1 — Синоним ELEMENT предоставляется, чтобы избежать двойного отрицания и для лучшего понимания текста. Обычно он будет использоваться (в противоположность команде кодирования ATTRIBUTE), для того чтобы различать природу типов высшего уровня в модуле ACH.1. Типы высшего уровня, которые не имеют окончательных команд кодирования ELEMENT и ATTRIBUTE, будут поддерживаться типами, которые прямо не соответствуют XML атрибутам или элементам и обычно бывают UNTAGGED.

Примечание 2 — Для команды кодирования ELEMENT не существует команды отрицания. ELEMENT может быть отменен последующей командой кодирования UNTAGGED, но такой прием не рекомендуется.

12.5 Каждое использование команды "PositiveInstruction" в префиксе типа XER или в управляющем разделе кодирования назначает эту команду кодирования XER соответствующему типу "Type".

12.6 Если "Type" в "TypeAssignment" (см. ИСО/МЭК 8824-1, 15.1) имеет окончательные команды кодирования, все применения "typereference" (в модуле, содержащем "TypeAssignment" или в некотором другом модуле) наследуют его окончательные ассоциированные команды кодирования, за исключением окончательных команд кодирования NAME и NAMESPACE (они не наследуются).

Примечание — Эти две команды кодирования влияют на имена XML, используемые вместо референтных имен типов. Если референтное имя типа используется для определения типа в назначении типа или типа компонента, наследование команд кодирования, по его определению, неуместно.

12.7 Команда кодирования в префиксе типа или в управляющем разделе кодирования может быть позитивной командой, используемой для того, чтобы добавить или заменить команду кодирования (использование "PositiveInstruction"), или негативной командой, используемой для отмены (использование "NegatingInstruction") одной или более команд кодирования.

12.8 Команды кодирования XER состоят из четырех частей (некоторые из которых могут быть пустыми):

а) NOT — показывает отрицание или удаление команд кодирования данной категории;

Примечание 1 — Эта часть присутствует для негативных команд (за исключением "ElementInstruction") и отсутствует для позитивных команд.

б) ключевое слово, идентифицирующее категорию команды кодирования;

Примечание 2 — Эта часть всегда присутствует.

в) идентификация целевого перечня назначений команд кодирования (возможно, с информацией для квалификации, ограничивающей его применение к подмножеству значений некоторого типа);

Примечание 3 — При использовании в префиксе типа целевой перечень всегда является "пустым" продукционным правилом, так как цель назначения — это всегда тип, ассоциированный с префиксом типа (см. 12.12). Целевой лист также всегда пустой для команды кодирования GLOBAL-DEFAULTS.

г) синтаксис, специфический для каждой категории команд кодирования, обеспечивающий детали кодирования в этой категории.

Примечание 4 — При использовании в негативной команде это всегда "пустое" продукционное правило. Эта часть всегда отсутствует в командах кодирования, для которых ключевое слово является достаточно обоснованным определением.

12.9 Некоторые команды кодирования XER требуют спецификации абстрактного значения типа. Эта спецификация использует продукционное правило "Value" (см. ИСО/МЭК 8824-1, 16.7). Если "valuereference" используется как "Value", то "valuereference" должно быть определено в (или импортировано в) модуле ACH.1, содержащем команду кодировании XER.

Примечание — Это означает, что значение может быть определено или непосредственно с применением базовой нотации значений ACH.1 или нотации значений XML.

12.10 В таблице 1 перечислены в первом столбце альтернативы в продукционных правилах "PositiveInstruction". Второй столбец содержит номера разделов, в которых определены требования для

применения тех команд кодирования, которые они продуцируют. В третьем столбце показаны категории каждой команды кодирования.

Примечание — Эти категории вводятся в таком порядке, чтобы обеспечить ясные утверждения о результате множественных назначений команд кодирования из той же самой категории.

Таблица 1 — Команды кодирования, разделы, где они определяются, и категории

Команда кодирования	Раздел определения	Категория
AnyAttributesInstruction	Раздел 17	Команда AnyAttributes
AnyElementInstruction	Раздел 18	Команда AnyElement
AttributeInstruction	Раздел 19	Команда Attribute
Base64Instruction	Раздел 20	Команда Base64
DecimalInstruction	Раздел 21	Команда Decimal
DefaultForEmptyInstruction	Раздел 22	Команда DefaultForEmpty
ElementInstruction	Раздел 23	Команда Element
EmbedValuesInstruction	Раздел 24	Команда EmbedValues
GlobalDefaultsInstruction	Раздел 25	Команда GlobalDefaults (но см. 14.3)
ListInstruction	Раздел 26	Команда List
NameInstruction	Раздел 27	Команда Name (но см. 14.3)
NamespaceInstruction	Раздел 28	Команда Namespace
PiOrCommentInstruction	Раздел 29	Команда PiOrComment (но см. 14.3)
TextInstruction	Раздел 30	Команда Text (но см. 14.3)
UntaggedInstruction	Раздел 31	Команда Untagged
UseNilInstruction	Раздел 32	Команда UseNil
UseNumberInstruction	Раздел 33	Команда UseNumber
UseOrderInstruction	Раздел 34	Команда UseOrder
UseQNameInstruction	Раздел 35	Команда UseQName
UseTypeInstruction	Раздел 36	Команда UseType
UseUnionInstruction	Раздел 37	Команда UseUnion
WhitespaceInstruction	Раздел 38	Команда Whitespace

12.11 Каждая из альтернатив для продукционного правила "PositiveInstruction" находится в определенной категории команды кодирования (или в некоторых случаях охватывает множественные категории). Категория каждой команды кодирования определена в столбце 3 таблицы 1 (но см. также 14.3 для команд кодирования, которые охватывают множественные категории).

Примечание — Категории команд кодирования используются в 14.4, чтобы определить эффект многократного назначения команд кодирования.

12.12 "TargetList" во всех конструкциях "EncodingInstruction", которые появляются в префиксе типа, будет "пуст", и адресатом должен быть "Type", связанный с префиксом типа.

12.13 Команда отрицания находится в той же категории, что и соответствующая положительная команда.

12.14 Тип ACH.1 никогда не мог связывать с этим больше чем одну команду кодирования XER данной категории (см. 14.3 и 14.4), независимо от того, как они назначены. Результат множественных назначений команды кодирования XER данной категории определен в разделе 14.

13 Назначение команды кодирования XER с использованием управляющих разделов кодирования

13.1 Перечень назначений команд кодирования

13.1.1 Команды кодирования XER могут быть назначены типам ACH.1 с использованием либо продукционного правила "EncodingInstruction" в префиксе типа XER, либо продукционного правила "EncodingInstructionAssignmentList" в разделе управления кодированием XER. Назначения, использующие префикс типа, определены в разделе 12. Назначения, использующие раздел управления кодированием XER, определены в этом разделе.

13.1.2 XER "EncodingInstruction" продукционное правило:

```
EncodingInstructionAssignmentList::=
        EncodingInstruction
        EncodingInstructionAssignmentList?
```

13.1.3 Продукционное правило "EncodingInstruction" определено в 13.3.

13.1.4 Каждое использование "EncodingInstruction" в разделе управления кодированием назначает эту команду кодирования XER появлению "Type", идентифицированному в команде кодирования "TargetList", или ссылкам на тип в перечне импорта. Продукционное правило "TargetList" и цели, которые идентифицируются перечнем, определены в 13.2.

13.1.5 Подразделы 12.4 и 12.14 также применяются к командам кодирования в разделе управления кодированием. Разделы, определяющие детализированный синтаксис для каждой категории команды кодирования, перечислены в таблице 1. Категории команд кодирования XER также перечислены в таблице 1.

13.2 Идентификация целей для команды XML кодирования с использованием перечня целей

13.2.1 Общие правила приведены ниже.

13.2.1.1 Альтернативы "EncodingInstruction" определяют команду кодирования XER, которая назначается, и цель (цели) для назначения в пределах модуля ACH.1. Все цели — появления продукционных правил "Type" в пределах модуля ACH.1.

Примечание — Могут быть определены множественные цели в тех же самых или в различных назначениях типа ACH.1. Цель, которая является полным модулем, или всеми появлениями в пределах модуля встроенного типа или конструктора, также может быть определена. Таким образом, (при использовании раздела управления XER кодирования) единственная "EncodingInstruction" может использоваться для назначения конкретной команды XER кодирования всем типам в модуле ACH.1, которым необходимо назначить эту команду кодирования.

13.2.1.2 При идентификации цели (целей) для назначения команды XER кодирования используется продукционное правило "TargetList". Это определено в следующих подразделах.

Примечание 1 — На продукционное правило "TargetList" ссылаются в разделе 19.

Примечание 2 — Продукционное правило "TargetList" имеет "пустую" альтернативу. Это — единственная разрешенная альтернатива, если "EncodingInstruction" используется в префиксе типа (см. 12.12). В 13.2 рассматривается только использование "TargetList" в управляющем разделе кодирования.

13.2.1.3 Продукционное правило "TargetList":

```
TargetList::=
        Targets "," +
        | empty
Targets::=
        TypeIdentification
        | BuiltInTypeIdentification
        | IdentifiersInContext
        | ImportedTypesIdentification
```

13.2.1.4 Если "TargetList" — перечень одного или более продукционных правил "Targets", то каждое из "Targets" идентифицирует одну или более целей (типы — "Type", которым назначена команда кодирования), но оно может также обеспечить уточняющую информацию для команды кодирования, ограничивая ее применение кодированием с использованием конкретного идентификатора в определении целевого типа, или с использованием тегов пустых элементов для управляющих символов, определенных в ИСО/МЭК 8824-1, 11.15.5.

Примечание — Уточняющая информация присутствует только в том случае, если цель является определением одного из следующих типов: булевский, строка битов, перечислимый, целочисленный или ограниченная строка символов (см. 13.2.2.9).

13.2.1.5 "TargetList" "пустых" разрешается только в префиксе типа (когда это — единственная разрешенная альтернатива) и в команде кодирования GLOBAL-DEFAULTS. В префиксе типа он идентифицирует тип, связанный с префиксом. В команде кодирования GLOBAL-DEFAULTS он идентифицирует весь "Type" в модуле.

13.2.1.6 Команда кодирования XER (возможно, со связанной уточняющей информацией) назначена всем типам, идентифицированным "TargetList", как определено в 13.2.1.10 — 13.2.1.16.

Примечание — Это было бы необычным, но правомочным для данного "Type", который будет идентифицирован не раз в целевом перечне. В таких случаях применяется раздел 14.

13.2.1.7 Идентификация цели (целей) (и, возможно, уточняющей информации) продукционным правилом "Targets" использует одну из пяти базовых форм:

а) использование ссылки на тип ("typereference", см. 13.2.2), возможно, сопровождаемой перечнем разделенных точкой идентификаторов, с распознаванием также:

1) "Type" в назначении типа (идентификаторы отсутствуют);

2) "Type" в компоненте определения типов (который может включать компоненты верхнего уровня, введенные конструкцией COMPONENTS OF — см. 13.2.1.12);

3) один из "Type" — (1) или (2), плюс конечный идентификатор, перед которым стоит двоеточие (не точка) для идентификатора, используемого в определении целевого типа, обеспечивает уточняющую информацию;

б) использование ALL как последнего идентификатора в форме (а), идентифицируя все, что типы ("Types") дословно представляют в определении типов (что идентифицировано предшествующей ссылкой на тип и перечнем идентификаторов, разделенных точками), или использование уточняющей информации (которой предшествует двоеточие, не точка), распознающей все идентификаторы, используемые для значений определений типов: булевских, строк битов, перечислимых, или целочисленных (которые идентифицированы предыдущей ссылкой на тип и списком идентификаторов, разделенных точками), или идентифицирующей все использования тегов пустого элемента XML, обычно представляющих некоторые управляющие символы (см. ИСО/МЭК 8824-1, 11.15.5);

в) использование "BuiltInTypeName" (см. 13.2.3), идентифицирующего все типы ("Type"s) в модуле, которые определены при помощи соответствующего встроенного названия типа или конструктора, возможно только в случае типов BOOLEAN, BIT STRING, ENUMERATED, INTEGER и ограниченной строки, в сопровождении уточняющей информацией;

г) использование списка "идентификаторов", сопровождаемого IN (или ALL сопровождаемого IN, или COMPONENTS, сопровождаемых IN) и формой (а), (см. также 13.2.4), идентифицирующего:

1) "Type" идентифицированных компонентов формы (а);

2) все "Types", которые текстуально встречаются в пределах "Type", идентифицированного формой (а) (использование ALL);

3) все ("Type"s), которые являются компонентами верхнего уровня "Type", идентифицированного формой (а) (использование COMPONENTS);

д) использование "ImportedTypesIdentification" (см. 13.2.5) идентифицирует все ссылки на типы ("typereference"s) в списке IMPORTS, которые импортированы из специфицированного модуля.

Примечание 1 — Термин "определение типов", используемый в формах (а) и (б) подчеркивает, что только дословно представленные идентификаторы могут использоваться. Идентификаторы не могут использоваться, если "Type" — ссылка на тип.

Примечание 2 — В общем случае на компонент можно сослаться при помощи форм (а) или (г). Если нужно сослаться больше чем на один компонент типа, то предпочтительнее вариант (г), так как это менее многословно, при других критериях надо отдать предпочтение форме (а). Это — вопрос стиля.

13.2.1.8 Тип строка битов или строка октетов с ограничением контента, который содержит тип, будет обработан как тип с единственным компонентом, с использованием "*" как составляющего идентификатора для назначения целевой команды типу ("Type") в ограничении содержания.

13.2.1.9 Определение типов, которые являются последовательностью-из или набором-из, должно быть обработано как тип с единственным компонентом, с использованием "*" как составляющего

идентификатора для назначения целевой команды типу ("Type"), который является компонентом последовательности-из или набора-из.

Примечание — Также возможно идентифицировать этот единственный компонент, используя составляющий идентификатор (если представлен).

13.2.1.10 Если цель — использование фиктивного параметра параметризованного типа, цель наследует окончательные команды кодирования фактического параметра прежде, чем команды кодирования, предназначенные для фиктивного параметра, назначены. Спецификация правомочна, если только результирующие окончательные команды кодирования для всех реализаций параметризованного типа правомочны.

Примечание 1 — Если параметризованный тип экспортирован, окончательные команды кодирования для его фиктивных параметров выполняются с ним.

Примечание 2 — Нет никаких механизмов, обеспечивающих назначение команды кодирования непосредственно типу ("Type") фактического параметра в реализации параметризованного типа.

13.2.1.11 Если цель — "SelectionType", она наследует окончательные команды кодирования выбранной альтернативы типа выбора, на который ссылается тип выбора, после которого назначаются команды кодирования, предназначенные типу "SelectionType".

13.2.1.12 Если цель — компонент, произведенный в результате COMPONENTS OF преобразования, она наследует окончательные команды кодирования компонента типа, на который ссылаются COMPONENTS OF. После преобразования назначаются команды кодирования, которые были назначены компонентам, произведенным COMPONENTS OF. Любые команды кодирования для типа "Type", из которого извлечены компоненты, игнорируются.

13.2.1.13 Если продукционное правило "Targets" — "TypeIdentification", то идентифицируемые цели определены в 13.2.2.

13.2.1.14 Если продукционное правило "Targets" — "BuiltInTypeIdentification", то идентифицируемые цели определены в 13.2.3.

13.2.1.15 Если продукционное правило "Targets" — "IdentifiersInContext", то идентифицируемые цели определены в 13.2.4.

13.2.1.16 Если продукционное правило "Targets" — "ImportedTypesIdentification", то идентифицируемые цели определены в 13.2.5.

13.2.1.17 Пример определения типов ACH.1, сопровождаемого двумя различными способами назначения команды кодирования XER в разделе управления кодированием, и то же самое определение типов ACH.1 с командами кодирования XER, назначенными с использованием префиксов типа. Все три подхода приводят к тому же самому результату EXTENDED-XER кодирования.

Определение типов:

```
My-Type::= SEQUENCE {
    field1 INTEGER,
    field2 CHOICE {
        first SEQUENCE OF INTEGER,
        second SEQUENCE OF OBJECT IDENTIFIER } }
```

Команды кодирования XER в разделе управления кодированием могли быть:

```
ATTRIBUTE field1 IN My-Type
LIST first IN My-Type.field2
```

Альтернативно, они могли быть:

```
ATTRIBUTE My-Type.field1
LIST My-Type.field2.first
```

Определение типов с префиксами типа:

```
My-Type::= SEQUENCE {
    field1 [ATTRIBUTE] INTEGER,
    field2 CHOICE {
        first [LIST] SEQUENCE OF INTEGER,
        second SEQUENCE OF OBJECT IDENTIFIER } }
```

13.2.2 Идентификация целей с использованием ссылки и идентификаторов типов ACH.1

13.2.2.1 Продукционное правило "TypeIdentification":

```
TypeIdentification ::=
        ALL
        | ModuleAndTypeReference
        ComponentReference ?
        QualifyingInformationPart ?
ModuleAndTypeReference ::=
        typereference
        | modulereference "." typereference
ComponentReference ::=
        "."
        ComponentIdList
ComponentIdList ::=
        ComponentId "." +
ComponentId ::=
        identifier
        | "*"
        | ALL
QualifyingInformationPart ::=
        ":"
        QualifyingInformation
        QualifyingInformation
        identifier
        | ALL
```

13.2.2.2 "TypeIdentification" ALL идентифицирует все типы ("Types") в назначениях типа ("TypeAssignment"s) в модуле.

13.2.2.3 Продукционное правило "ModuleAndTypeReference" идентифицирует "Type", который назначен "typereference". "Modulereference" в "ModuleAndTypeReference" должен быть ссылкой на модуль для модуля, содержащего "EncodingInstructionAssignmentList", и "typereference" должен быть ссылкой на тип, который определен в модуле. Это должно использоваться, если и только если "typereference" состоит из тех же самых символов как одно из ключевых слов, определенных в 11.3, иначе одна ссылка "typereference" должна быть использована.

13.2.2.4 Символ "*" идентифицирует "Type" (единственного) компонента типа последовательность-из или набора-из, или типа в ограничении контента, которое содержит "Type".

Примечание — Эта форма может использоваться, даже если компонент типа последовательность-из или набор-из имеет идентификатор, но использование идентификатора должно быть предпочтительней.

13.2.2.5 Если ключевое слово ALL будет использоваться как "ComponentId", то это должен быть последний "ComponentId" в "ComponentIdList" и он не должен сопровождаться "QualifyingInformation".

13.2.2.6 Если первый "ComponentId" в "ComponentIdList" (если присутствует) — идентификатор, который дословно присутствует (или следует из использования COMPONENTS OF) как составляющий идентификатор в "Type", идентифицированном "ModuleAndTypeReference", то это идентифицирует "Type" того компонента. Если это не идентификатор, который дословно присутствует (или следует из использования COMPONENTS OF) как составляющий идентификатор в "Type", идентифицированном "ModuleAndTypeReference", то это появление "TypeIdentification" не правомочно, кроме того, не идентифицирует никакой цели.

Примечание — Это требует, чтобы тип, на который ссылается "ModuleAndTypeReference", был определением типов: последовательность, набор, выбор, последовательность-из, или набор-из, или определением типов: строка битов или строка октетов с ограничением содержания, которое содержит "Тип".

13.2.2.7 Если последующий "ComponentId" (кроме последнего) в "ComponentIdList" (если присутствует) — идентификатор, который дословно присутствует как идентификатор компонента в "Type", идентифицированном предыдущим "ComponentId", то идентифицируется "Type" того компонента. Если это не идентификатор компонента, который дословно присутствует в "Type", идентифицированном предыдущим "ComponentId", то появление "TypeIdentification" не правомочно, кроме того, не идентифицирует никакой цели.

Примечание — В первом использовании "ComponentId" можно ссылаться на компоненты, введенные COMPONENTS OF. Компоненты этих компонентов не могут быть идентифицированы последующим "ComponentId".

13.2.2.8 Если последний "ComponentId" в "ComponentIdList" (если присутствует):

а) идентификатор, который дословно присутствует как идентификатор компонента в "Type", идентифицированном предыдущим "ComponentId", то он идентифицирует "Type" этого компонента, команда кодирования должна быть назначена этому "Type";

б) ключевое слово ALL — команда кодирования должна быть назначена всем типам "Types", которые дословно присутствуют в определении типов, идентифицированном предыдущим "ComponentId", который должен быть типом с одним компонентом или более.

13.2.2.9 "QualifyingInformationPart" не должна присутствовать, если только "ModuleAndTypeReference" с "ComponentReference" (если присутствует) не идентифицирует цели:

а) булевского типа;

б) типа строки битов с поименованными битами;

в) перечислимого типа;

г) целочисленного типа с поименованными числами;

д) типа строки с ограниченным набором символов.

13.2.2.10 Альтернатива "identifier" для "QualifyingInformation" не должна использоваться, если "ModuleAndTypeReference" с "ComponentReference" (если присутствует) не идентифицирует единственной цели, которая не принадлежит к типу строки с ограниченным набором символов, или идентифицирует перечень целей булевского типа. "Идентификатор" должен быть идентификатором в определении типа цели, если цель не принадлежит к булевскому типу, или должен быть true или false. "Идентификатор" является уточняющей информацией, которая распознает, что команда кодирования применяется только к кодированию с использованием того идентификатора.

13.2.2.11 Альтернативы true и false для "QualifyingInformation" для булевского типа определяют уточняющую информацию, которая идентифицирует, что команда кодирования применяется только к кодированию абстрактных значений true или false соответственно.

13.2.2.12 Альтернатива ALL для "QualifyingInformation" не должна использоваться, если цель не идентифицирует (только) одно или более определений типов для типов, перечисленных в 13.2.2.9. Она не должна использоваться, если цель идентифицирует одну или более целей типа строки с ограниченным набором символов, если применяемая команда кодирования не NAMESPACE. Она определяет уточняющую информацию, которая идентифицирует положение, что команда кодирования применяется ко всем идентификаторам в определениях типов, или в случае типа строки ограниченного набора символов — ко всем вариантам использования тегов пустых элементов XML, применяемых для представления управляющих символов, перечисленных в ИСО/МЭК 8824-1, 11.15.5.

Примечание — Невозможно использовать уточняющую информацию с "идентификатором", чтобы выборочно влиять на представление управляющих символов. Только ALL возможно в этом случае.

13.2.3 Целевые идентификации при использовании имени встроенного типа

13.2.3.1 Продукционное правило "BuiltInTypeIdentification":

```
BuiltInTypeIdentification::=
        BuiltInTypeName
        BuiltInTypeQualifyingInformationPart ?
BuiltInTypeName::=
        BIT STRING
        |BOOLEAN
        | CHARACTER STRING
        | CHOICE
        | EMBEDDED PDV
        | ENUMERATED
        | EXTERNAL
        | GeneralizedTime
        | INSTANCE OF
        | INTEGER
        | NULL
        | ObjectDescriptor
        | OBJECT IDENTIFIER
```

| OCTET STRING
| REAL
| RELATIVE-OID
| SEQUENCE
| SEQUENCE OF
| SET
| SET OF
| UTCTime
| RestrictedCharacterStringType

BuiltInTypeQualifyingInformationPart::=
":"
BuiltInTypeQualifyingInformation

BuiltInTypeQualifyingInformation
identifier
| ALL

13.2.3.2 Продукционное правило "BuiltInTypeIdentification" определяет, что команда кодирования должна быть применена ко всем дословным появлениям в пределах модуля соответствующего встроенного типа или типа, определенного с использованием соответствующего конструктора.

13.2.3.3 "RestrictedCharacterStringType" определен в ИСО/МЭК 8824-1, 37.

13.2.3.4 "BuiltInTypeQualifyingInformationPart" не должна присутствовать, если только "BuiltInTypeName" не принадлежит типам: булевский, строка битов, перечислимый, целочисленный, или строка ограниченного набора символов.

Примечание — Только форма ALL для "BuiltInTypeQualifyingInformation" разрешается для строки ограниченного набора символов (см. 13.2.2.10 и следующие подразделы).

13.2.3.5 Альтернатива "identifier" для "BuiltInTypeQualifyingInformation" не должна использоваться, если "BuiltInTypeName" не булевского типа, и тогда значение должно быть true или false. Это определяет уточняющую информацию, идентифицирующую команду кодирования как применяемую только к кодированию абстрактных значений true или false, соответственно.

13.2.3.6 Альтернатива ALL для "BuiltInQualifyingInformation" определяет уточняющую информацию, которая идентифицирует команду кодирования как применяемую ко всем идентификаторам в любом случае использования "BuiltInTypeName" в пределах модуля (или ко всем значениям определения булевского типа, или ко всем тегам пустых элементов, используемым в значениях специфицированного типа строки ограниченного набора символов — см. ИСО/МЭК 8824-1, 11.15.5).

13.2.4 Использование идентификаторов в контексте

13.2.4.1 Продукционное правило "IdentifiersInContext":

IdentifiersInContext ::=
IdentifierList
IN
TypeIdentification

IdentifierList ::=
identifier "," +
| ALL
| COMPONENTS

13.2.4.2 "TypeIdentification" определена в 13.2.2 и идентифицирует тип, определенный в операторе присваивания типа в модуле или компоненте, или субкомпоненте типа, определенного в модуле. "QualifyingInformationPart" должна отсутствовать.

13.2.4.3 "Type", идентифицированный "TypeIdentification", должен быть одним из типов: последовательность, набор или выбор, и называться в целях этого раздела идентифицированным "Type".

Примечание — "TypeIdentification" в "IdentifiersInContext" не может использоваться для типов последовательность-из или набор-из. Такое использование запрещено для ясности, поскольку это было бы не менее многословным, чем прямое использование "TypeIdentification" в "Targets".

13.2.4.4 Каждый "identifier" в "IdentifierList" должен быть идентификатором компонента, идентифицированного "Type". Команда кодирования XER назначена типу всех компонентов идентифицированного "Type", которые имеют идентификатор компонента в "IdentifierList".

13.2.4.5 Использование ALL для "IdentifierList" определяет, что все дословно существующие компоненты (и все дословно существующие компоненты тех компонентов до любой глубины) в идентифицированном "Type" — цели, которым назначается команда кодирования XER.

13.2.4.6 Использование COMPONENTS для "IdentifierList" определяет, что все компоненты (на первом уровне) идентифицированного "Type" — цели, которым назначается команда кодирования XER.

13.2.5 Использование идентификации импортированных типов.

13.2.5.1 Продукционное правило "ImportedTypesIdentification":

ImportedTypesIdentification::=

ALL IMPORTS FROM modulereference

13.2.5.2 "Modulereference" должна быть одной из ссылок "modulereference", используемых в одной из "GlobalModuleReferences" раздела по импорту модуля.

13.2.5.3 Команда кодирования XER назначена каждой из "typereference" в соответствующем "SymbolList" после того, как окончательные команды кодирования, продуцированные назначением в модуле экспорта, были назначены.

13.2.5.4 Если импортированная "typereference" экспортируется из этого модуля, окончательные команды кодирования, унаследованные этой "typereference" в модуле, который импортирует ее, — унаследованные в этом модуле импортирования, и не затронуты назначением команд кодирования, с использованием "ImportedTypesIdentification". Это назначение затрагивает только использование ссылки на тип в пределах этого модуля.

14 Множественное назначение команд кодирования XER

14.1 Порядок, в котором рассматривается множественное назначение

14.1.1 "Type", который не является ссылкой на тип ("typereference"), имеет первоначально пустой набор ассоциированных команд кодирования.

14.1.2 "Type", который является ссылкой на тип ("typereference") (который может быть импортирован) имеет первоначально набор окончательных команд кодирования "Type", назначенный этому набору, когда он был определен (возможно, измененный командами кодирования, назначенными ему в списке импорта модуля импортирования — см. 13.2.5).

14.1.3 Целевые команды кодирования для "Type" (при использовании раздела управления кодированием) назначаются следующими в порядке, в котором целевые команды кодирования появляются в разделе управления кодированием. Если "Type" идентифицирован более чем одним элементом "TargetList" (см. 13.2), то должна быть обработка как многократных назначений той же самой команды кодирования к тому "Type" в порядке, в котором элементы встречаются в "TargetList".

Примечание — Эффект 14.1.2 и 14.1.3 означает, что целевое назначение типу "Type" в "TypeAssignment" всегда отменяется целевым назначением типу "Type", определенному с использованием соответствующей ссылки "typereference", независимо от того, какое целевое назначение появится первым в разделе управления кодированием. Однако, если целевое назначение сделано всем компонентам типа, и также некоторому индивидуальному компоненту этого типа, то эффект будет зависеть от порядка команд кодирования в разделе управления кодированием.

14.1.4 Снабженные префиксами команды кодирования (использующие префикс типа), назначенные типу, рассматриваются следующим образом: крайняя правая (самая внутренняя) команда кодирования, снабженная префиксом рассматривается первой, а крайняя левая (наиболее удаленная) команда кодирования, снабженная префиксом, рассматривается последней.

14.1.5 Как определено в 13.2.1.10, команды кодирования назначаются фиктивному параметру только после того, как окончательные команды кодирования для фактического параметра были определены.

14.1.6 Как определено в 13.2.1.11 и 13.2.1.12, "Тип выбора" ("SelectionType") и компоненты, продуцированные преобразованием COMPONENTS OF, наследуют сначала окончательные команды кодирования оригинального типа, и затем применяют команды кодирования, предназначенные для них.

14.1.7 Каждое назначение команды кодирования продуцирует новый набор связанных команд кодирования, как определено в 14.2 — 14.4.

14.2 Эффект назначения команды кодирования отрицания

14.2.1 Все назначения команды кодирования отрицания приводят к удалению (из набора связанных команд кодирования) любой команды кодирования той же самой категории. Если нет никаких связанных команд кодирования другой категории, набор становится пустым.

14.2.2 Команда кодирования NOT GLOBAL-DEFAULTS никогда не должна назначаться.

14.2.3 Для команд кодирования с множественными категориями (см. 14.3), команда кодирования отрицания удаляет все команды кодирования в любой из категорий.

Примечание — Команда кодирования отрицания никогда не становится частью набора связанных команд кодирования.

14.3 Множественное назначение команд кодирования с множественными категориями

14.3.1 Команды кодирования NAME и TEXT (см. разделы 27 и 30) могут быть назначены типу с целью:

а) изменить ассоциированное имя тега (нет никакой уточняющей информации "QualifyingInformation");

Примечание — Это применяется только к команде кодирования NAME.

б) изменить результат кодирования "ExtendedXMLValue", обеспечивая новое название для специфицированного идентификатора "identifier", присутствующего в определении типа (присутствует уточняющая информация "QualifyingInformation", которая не является ALL);

в) изменить результат кодирования "ExtendedXMLValue", обеспечив модификацию, которая будет применена ко всем идентификаторам "identifiers", присутствующим в определении типов (присутствует уточняющая информация "QualifyingInformation", которая является ALL, с целью, которая не является типом ограниченной строки символов).

14.3.2 В случае 14.3.1 (б) команда кодирования для специфицированного идентификатора "identifier" обрабатывается как категория, отличная от команды кодирования для любого другого идентификатора "identifier", и от команды кодирования для 14.3.1 (а).

14.3.3 В случае 14.3.1 (в) команда кодирования попадает в набор команд кодирования типа 14.3.1 (б) с одной командой кодирования для каждого идентификатора "identifier", присутствующего в определении типов.

14.3.4 Команда кодирования PI-OR-COMMENT (см. раздел 29) имеет четыре категории, соответствующие четырем альтернативам для "Position".

14.3.5 В соответствии с 14.3.3 и 14.3.4, подраздел 14.4 определяет правила для множественного назначения команд кодирования XER.

14.3.6 Каждая из альтернатив команды кодирования GLOBAL-DEFAULTS — отдельная категория, но каждая категория этой команды кодирования должна быть назначена не больше одного раза.

14.4 Множественное назначение команд кодирования XER той же категории

Примечание — Предполагается, что множественное назначение команд кодирования XER той же категории будет редким, за исключением тех случаев, когда команда кодирования XML назначается глобально, и команда кодирования замены (возможно, отрицания) назначена определенным типам или компонентам. Этот подраздел определяет правила для случев, когда множественное назначение команд кодирования XER происходит в той же самой категории. На этот раздел также ссылается пункт 14.3.5 для обработки множественных назначений команд кодирования NAME, PI-OR-COMMENT и TEXT.

14.4.1 Назначение положительных команд кодирования приводит к добавлению (к набору связанных команд кодирования) этой команды кодирования XER, если нет никаких других связанных команд кодирования той же категории.

14.4.2 Назначение команды кодирования ELEMENT всегда эквивалентно назначению NOT UNTAGGED команды кодирования.

14.4.3 Если есть команда кодирования той же самой категории в наборе связанных команд кодирования, то эта команда кодирования удаляется из набора, а назначенная команда кодирования XER добавляется.

Примечание — Если команды кодирования назначаются глобально в раздел управления кодированием с намерением отменить их в определенных случаях, то отмена должна быть сделана с использованием или префикса типа или более поздней команды кодирования в разделе управления кодированием, но не более ранней.

14.4.4 Если тип, который появляется либо в "ОграниченииКонтента" ("ContentsConstrait"), либо в "ОграниченииТипа" ("TypeConstraint") должен быть закодирован по правилам EXTENDED-XER, то окончательные команды кодирования (как определено правилами выше) используются при определении кодирования этого типа. Если тип появляется в любом другом ограничении АСН.1, то все связанные команды кодирования не обрабатываются

14.5 Разрешенные комбинации окончательных команд кодирования

14.5.1 Таблица 2 определяет разрешенные комбинации окончательных команд кодирования для "Типа" ("Type"), когда использовалась GLOBAL-DEFAULTS of MODIFIED-ENCODINGS. В графе 1 перечислены все команды кодирования, в графе 2 перечислены все команды кодирования, которые могут использоваться в комбинации с командой кодирования графы 1 как окончательной командой кодирования. Но во многих случаях применяются ограничения, которые перечислены в соответствующих разделах.

Примечание — GLOBAL-DEFAULTS не приводится в таблице, поскольку она не назначается типу.

Таблица 2 — Разрешенные комбинации окончательных команд кодирования с MODIFIED-ENCODINGS

Команда кодирования	Другие разрешенные команды кодирования
ANY-ATTRIBUTES *(см. раздел 17)*	**ELEMENT, NAME, NAMESPACE**
ANY-ELEMENT *(см. раздел 18)*	**ELEMENT, NAME, NAMESPACE**
ATTRIBUTE *(см. раздел 19)*	**BASE64, DECIMAL, ELEMENT, LIST, NAME, NAMESPACE, TEXT, USE-NUMBER, USE-QNAME, USE-UNION, WHITESPACE**
BASE64 *(см. раздел 20)*	**ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED**
DECIMAL *(см. раздел 21)*	**ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED**
DEFAULT-FOR-EMPTY *(см. раздел 22)*	**BASE64, DECIMAL, ELEMENT, EMBED-VALUES, LIST, NAME, NAMESPACE, PI-OR-COMMENT, TEXT, USE-NIL, USE-NUMBER, USE-ORDER, USE-QNAME, USE-UNION, WHITESPACE**
ELEMENT *(см. раздел 23)*	Эквивалентный **NOT UNTAGGED**
EMBED-VALUES *(см. раздел 24)*	**DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, USE-NIL, USE-ORDER**
LIST *(см. раздел 26)*	**ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED**
NAME *(см. раздел 27)*	Никаких ограничений
NAMESPACE *(см. раздел 28)*	Никаких ограничений
PI-OR-COMMENT *(см. раздел 29)*	**BASE64, DECIMAL, DEFAULT-FOR-EMPTY, ELEMENT, EMBEDVALUES, LIST, NAME, NAMESPACE, TEXT, USE-NIL, USE-NUMBER, USE-ORDER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE**
TEXT *(см. раздел 30)*	**ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED**
UNTAGGED *(см. раздел 31)*	**BASE64, DECIMAL, LIST, NAME, NAMESPACE, TEXT, USENUMBER, USE-QNAME, USE-UNION, WHITESPACE**
USE-NIL *(см. раздел 32)*	**DEFAULT-FOR-EMPTY, ELEMENT, EMBED-VALUES, NAME, NAMESPACE, PI-OR-COMMENT, USE-ORDER**
USE-NUMBER *(см. раздел 33)*	**ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED**
USE-ORDER *(см. раздел 34)*	**DEFAULT-FOR-EMPTY, ELEMENT, EMBED-VALUES, NAME, NAMESPACE, PI-OR-COMMENT, USE-NIL.**

Окончание таблицы 2

Команда кодирования	Другие разрешенные команды кодирования
USE-QNAME *(см. раздел 35)*	**ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED**
USE-TYPE (*см. раздел 36)*	**ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT**
USE-UNION *(см. раздел 37*)	**ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED**
WHITESPACE *(см. раздел 38)*	**ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED**

14.5.2 Таблица 3 определяет разрешенные комбинации окончательных команд кодирования, когда команда GLOBAL-DEFAULTS MODIFIED-ENCODINGS не использовалась. В графе 1 перечислены все команды кодирования, которые разрешаются как окончательные команды кодирования, если команда GLOBAL-DEFAULTS MODIFIED-ENCODINGS не использовалась. В графе 2 приведены "Не разрешено" или перечислены все команды кодирования, которые могут использоваться в комбинации с командой кодирования графы 1, как окончательной командой кодирования, но во многих случаях применяются ограничения, которые перечислены в соответствующих разделах. "Не разрешено" означает, что команда кодирования не может использоваться как окончательная команда кодирования, если команда GLOBAL-DEFAULTS MODIFIED-ENCODINGS не использовалась.

Примечание — GLOBAL-DEFAULTS не приведена в таблице, поскольку она не назначается "Типу" ("Type").

Таблица 3 — Разрешенные комбинации окончательных команд кодирования без MODIFIED-ENCODINGS

Команда кодирования	Другие разрешенные команды кодирования
ANY-ATTRIBUTES	Не разрешено
ANY-ELEMENT	Не разрешено
ATTRIBUTE	**BASE64, LIST, NAME, TEXT, USE-NUMBER, WHITESPACE**
BASE64	**ATTRIBUTE, NAME, PI-OR-COMMENT**
DECIMAL	Не разрешено
DEFAULT-FOR-EMPTY	Не разрешено
ELEMENT	Не разрешено
EMBED-VALUES	Не разрешено
LIST	**ATTRIBUTE, NAME, PI-OR-COMMENT**
NAME	**ATTRIBUTE, BASE64, LIST, PI-OR-COMMENT, TEXT, USENUMBER, WHITESPACE**
NAMESPACE	Не разрешено
PI-OR-COMMENT	**BASE64, LIST, NAME, TEXT, USE-NUMBER, WHITESPACE**
TEXT	**ATTRIBUTE, NAME, PI-OR-COMMENT**
UNTAGGED	Не разрешено
USE-NIL	Не разрешено
USE-NUMBER	**ATTRIBUTE, NAME, PI-OR-COMMENT**
USE-ORDER	Не разрешено
USE-QNAME	Не разрешено

Окончание таблицы 3

Команда кодирования	Другие разрешенные команды кодирования
USE-TYPE	Не разрешено
USE-UNION	Не разрешено
WHITESPACE	**ATTRIBUTE, NAME, PI-OR-COMMENT**

15 Поддержка команд кодирования XER для пространства имен и уточненных имен XML

15.1 Рекомендация W3C XML Namespaces определяет понятия и правила, управляющие необходимыми классификаторами и механизмами, чтобы гарантировать, что имя элемента XML или атрибута могут быть правильно идентифицированы с соответствующей спецификацией связанной семантики.

15.2 W3C XML Namespaces определяет пространства имен XML как набор однозначных имен, идентифицированных URI, которые используются в документах XML в качестве типов элементов и имен атрибутов. URI, которое идентифицирует пространство имен, называется именем пространства имен. В настоящем стандарте пространство имен также используется, чтобы классифицировать значения типа, который имеет окончательную команду кодирования USE-QNAME (см. 35) и который представляет XML QName (см. W3C XML Схема, Часть 2, 3.2.18).

15.3 Имена ссылок на тип и идентификаторы могут (но не обязательно) быть назначенными пространству имен.

Примечание — Настоящий стандарт использует имя пространства имен, т.е., по умолчанию, форму URI, основанную на идентификаторах объекта ACH.1 (см. раздел 28). Все другие формы URI могут использоваться, чтобы назначить имя пространства имен именам в модуле ACH.1.

15.4 Принадлежность типа части пространства имен (и если так, его имени пространства имен) определяется присутствием (или отсутствием) окончательной команды кодирования NAMESPACE.

Примечание — Команда кодирования NAMESPACE может присутствовать, только если команда кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS также присутствует в разделе управления кодированием (см. 28.2.1).

15.5 Пространство имен идентифицировано продукционным правилом "NamespaceSpecification", которое обеспечивает Унифицированный Идентификатор Ресурса для пространства имен, и опционально рекомендованный префикс пространства имен. "NamespaceSpecification" определена в разделе 28.

15.6 Имена элементов и атрибутов XML при EXTENDED-XER кодировании генерируются из нескольких источников. В 15.8 перечислены источники имен элементов и атрибутов XML, идентифицировано пространство имен, частью которого они являются, и определено, должны ли они быть именами в уточненном пространстве имен или нет.

15.7 Имя элемента XML, имя атрибута XML или значение атрибута идентификации типа могут (но не обязательно) иметь окончательную команду кодирования NAMESPACE, которая, применяется к "Типу" ("Type") и генерирует имя. Если происходит это генерирование, имя должно быть именем в уточненном пространстве имен при кодировании (уточненное пространство имен при кодировании может быть создано или явно с использованием определенного префикса пространства имен XML, или неявно путем установления пространства имен XML для диапазона действий, который включает использование имени или значения). Если нет команды кодирования NAMESPACE, примененной к "Типу" ("Type"), которая генерирует имя, тогда имя не является именем уточненного пространства имен. Имена, которые не являются именами уточненного пространства имен, называются неуточненными именами, и не должны встречаться в пределах установленного значения по умолчанию пространства имен XML.

Примечание — Базовые правила BASIC-XER не поддерживают пространство имен XML, и имена уточненного пространства имен никогда не встречаются при кодировании.

15.8 В следующих подразделах термин "Пространство имен ACH.1" относится к пространству имен, чье имя и рекомендованный префикс определены в 15.9. Термин "назначенное пространство имен" относится к пространству имен, назначенному командой кодирования NAMESPACE типу. Если сгенерированные имена не из пространства имен ACH.1 и нет такого назначения пространства имен, то имена элементов XML, атрибутов XML и значения атрибутов идентификации типа являются неуточненными именами.

15.8.1 Во всех подпунктах подраздела 15.8 имена элементов и атрибутов в тегах XML (будут ли теги тегами пустого элемента или начальными тегами) являются именами уточненного пространства имен при кодировании, если и только если порождающий "Тип" ("Type") имеет окончательную команду кодирования NAMESPACE.

15.8.2 Названия элементов в тегах пустого элемента XML, используемых для управляющих символов (см. ИСО/МЭК 8824-1, 11.15.5), не имеют пространства имен, пока одно не будет назначено применением команды кодирования NAMESPACE к типу "ограниченная строка символов" с уточняющей информацией ALL.

15.8.3 Имена элементов в тегах пустого элемента XML, используемых для значений типов целый, перечислимый, строка битов и специальных значений действительных типов (см. ИСО/МЭК 8824-1, 18.9, 19.8, 20.6 и 21.9), будут всегда неуточненными именами (см. 15.7) при кодировании этих типов.

15.9 Пространство имен атрибута идентификации типа (см. раздел 36) и атрибута идентификации пустого множества (см. раздел 32) — это управляющее пространство имен, которое, по умолчанию, является пространством имен ACH.1, если только другое управляющее пространство имен не определено командой кодирования GLOBAL-DEFAULTS (см. раздел 25). Пространство имен ACH.1 имеет имя **"urn:oid:2.1.5.2.0.1"** (см. 39.3), и рекомендованный префикс пространства имен **"asn.1"** (см. также 25.3.2.).

15.10 Для типа строка октетов с ограничением содержания, которое определяет EXTENDED-XER кодирование, любое абстрактное значение типа строка октетов должно быть полным EXTENDED-XER кодированием значения типа ACH.1 (см. ИСО/МЭК 8824-3, 11.5 и 11.6) и будет содержать все необходимые декларации пространства имен для всех уточненных имен с префиксом и без префикса, присутствующих в абстрактном значении строки октетов.

Примечание — Такой тип (строка октетов) закодирован как "xmlhstring" или "Base64OctetstringValue". Любые объявления пространства имен, представленные в документе XML, который содержит "xmlhstring", или "Base64OctetstringValue" не включают в их область применения имена, представленные в строке октетов.

15.11 Когда открытый тип закодирован как "xmlhstring" или "Base64XMLOpenTypeFieldVal" и правила кодирования, используемые для содержавшегося типа, являются EXTENDED-XER, "xmlhstring" или "Base64XMLOpenTypeFieldVal" должны иметь шестнадцатеричное или base64 представление (соответственно) строки октетов, которая является полным EXTENDED-XER кодированием значения содержавшегося типа, и должна содержать все необходимые декларации пространства имен для всех имен с префиксом или без префикса, представленных в ней.

Примечание — Любые декларации пространства имен, представленные в документе XML, который содержит ' 'xmlhstring" или "Base64XMLOpenTypeFieldVal" не включают в их область применения имена, представленные в строке октетов.

16 Спецификации кодирования EXTENDED-XER

Спецификация кодирования EXTENDED-XER использует продукционные правила, определенные в следующих подразделах. Эти правила позволяют применять весь синтаксис соответствующих продукционных правил, используемых BASIC-XER (того же самого названия, но с удаленным "Extended"), но обеспечивают дополнительный синтаксис, который допускается в кодировании EXTENDED-XER. Использование этого дополнительного синтаксиса определено применением команд кодирования XER (см. 17 — 38).

Примечание — Альтернативные доступные продукционные правила часто ограничиваются использованием или не использованием команды кодирования GLOBAL-DEFAULTS с ключевым словом MODIFIED-ENCODINGS (см. 9.2.7 и 9.2.8). В частности, таким ограничением управляется использование пустого элемента или текстового кодирования для некоторых встроенных типов.

16.1 Элемент XML документа

16.1.1 Элемент XML документа должен быть "ExtendedXMLTypedValue".

16.1.2 "ExtendedXMLTypedValue" — это:

```
ExtendedXMLTypedValue::=
        "<" & TypeNameOrModifiedTypeName AttributeList ">"
ExtendedXMLValue
        "</" & TypeNameOrModifiedTypeName ">"
        | "<" & TypeNameOrModifiedTypeName "/>"
```

Примечание — Отличие от продукционного правила "XMLTypedValue" — включение возможно пустого "AttributeList" и использование "ExtendedXMLValue" вместо "XMLValue" для содержания элемента XML.

16.1.3 "TypeNameOrModifiedTypeName" определено в 16.2.

16.1.4 "AttributeList" определен в 16.3.

16.1.5 "ExtendedXMLValue" определено в 16.4 и должно быть типа "ExtendedXMLValue", идентифицированного "TypeNameOrModifiedTypeName".

16.1.6 Вторая альтернатива "XMLTypedValue" (использование тега пустого элемента XML) может использоваться, только если продукционное правило "ExtendedXMLValue" является пустым.

Примечание — Если бы продукционное правило "ExtendedXMLValue" было "xmlcstring", содержащим только "пробел", оно не было бы пустым, и вторая альтернатива не могла бы использоваться.

16.2 Продукционное правило "TypeNameOrModifiedTypeName"

16.2.1 "TypeNameOrModifiedTypeName" — это:

```
TypeNameOrModifiedTypeName::=
        NonParameterizedTypeName
        | QualifiedOrUnqualifiedName
```

16.2.2 Непараметризованное имя типа "NonParameterizedTypeName" определено в ИСО/МЭК 8824-1, 13.2, и используется (как определено в том подразделе и в ИСО/МЭК 8824-1, 13.4–13.7) как имя элемента XML, которое идентифицирует тип ACH.1.

16.2.3 Уточненное или неуточненное имя "QualifiedOrUnqualifiedName" определено в 28.3.2. Альтернатива "QualifiedOrUnqualifiedName" должна использоваться, если и только если есть окончательная команда кодирования NAME или NAMESPACE, примененная к типу (см. 27), иначе должно использоваться непараметризованное имя типа "NonParameterizedTypeName".

16.3 Продукционное правило "AttributeList"

16.3.1 "AttributeList" — это:

```
AttributeList::=
Attribute AttributeList
        | empty
```

16.3.2 "Attribute" определен в 19.3.3.

16.3.3 "AttributeList" будет пуст, если применение окончательных команд кодирования не требует его использования (см. разделы 19, 32 и 36).

16.3.4 Атрибутам ("Attribute") в "AttributeList" будет предшествовать "пробел" (см. 7.1.4).

16.4 Продукционное правило "ExtendedXMLValue"

16.4.1 "ExtendedXMLValue":

```
ExtendedXMLValue::=
        ExtendedXMLBuiltinValue
        | ExtendedXMLObjectClassFieldValue
        | empty
ExtendedXMLBuiltinValue::=
        XMLBitStringValue
        | XMLBooleanValue
        | ExtendedXMLCharacterStringValue
        | ExtendedXMLChoiceValue
        | XMLEmbeddedPDVValue
        | ExtendedXMLEnumeratedValue
        | XMLExternalValue
        | XMLInstanceOfValue
        | ExtendedXMLIntegerValue
        | XMLNullValue
        | XMLObjectIdentifierValue
        | ExtendedXMLOctetStringValue
        | ExtendedXMLRealValue
        | XMLRelativeOIDValue
        | ExtendedXMLSequenceValue
        | ExtendedXMLSequenceOfValue
        | ExtendedXMLSetValue
        | ExtendedXMLSetOfValue
        | ExtendedXMLPrefixedValue
```

ExtendedXMLCharacterStringValue::=
ExtendedXMLRestrictedCharacterStringValue
| XMLUnrestrictedCharacterStringValue
ExtendedXMLRestrictedCharacterStringValue::=
XMLRestrictedCharacterStringValue
| Base64XMLRestrictedCharacterStringValue
ExtendedXMLObjectClassFieldValue::=
ExtendedXMLOpenTypeFieldVal
| XMLFixedTypeFieldVal
ExtendedXMLOpenTypeFieldVal::=
ExtendedXMLTypedValue
| Base64XMLOpenTypeFieldVal
| xmlhstring
ExtendedXMLOctetStringValue::=
ExtendedXMLTypedValue
| Base64XMLOctetStringValue
| xmlhstring
ExtendedXMLRealValue::=
XMLRealValue
| ModifiedXMLRealValue
ExtendedXMLIntegerValue::=
XMLIntegerValue
| ModifiedXMLIntegerValue
ExtendedXMLPrefixedValue::=
ExtendedXMLValue

16.4.2 Альтернативы "ExtendedXMLBuiltinValue", чьи имена продукционных правил не начинаются с "Extended", их использование для кодирования абстрактных значений полностью определены в ИСО/МЭК 8824-1 (см. 15.10 и 15.2 настоящего стандарта) и (для "XMLFixedTypeFieldVal" и третьей альтернативы для "ExtendedXMLOpenTypeFieldVal") в ИСО/МЭК 8824-2, 14.6.

16.4.3 "Base64XMLRestrictedCharacterStringValue" определено в 20.3.5 и должно использоваться только так, как определено в этом подразделе.

16.4.4 "ExtendedXMLChoiceValue" определено в 16.5 и должно использоваться только так, как определено в этом подразделе.

16.4.5 "ExtendedXMLEnumeratedValue" определено в 33.3 и должно использоваться только так, как определено в этом подразделе.

16.4.6 "ExtendedXMLSequenceValue" и "ExtendedXMLSetValue" определены в 16.6 и должны использоваться только так, как определено в этом подразделе.

16.4.7 "ExtendedXMLSequenceOfValue" и "ExtendedXMLSetOfValue" определены в 16.7 и должны использоваться только так, как определено в этом подразделе.

16.4.8 "Base64XMLOctetStringValue" и "Base64XMLOpenTypeFieldVal" определены в 20.3.2 и 20.3.4 и должны использоваться только так, как определено в этих подразделах.

16.4.9 "ModifiedXMLIntegerValue" определено в 16.8 и должно использоваться только так, как определено в этом подразделе.

16.4.10 "ModifiedXMLRealValue" определено в 16.9 и должно использоваться только так, как определено в этом подразделе.

16.4.11 "Пустая" альтернатива для "ExtendedXMLValue" должна использоваться только так, как определено в разделе 22.

Примечание — Другие альтернативы для "ExtendedXMLValue" могут также продуцировать "пустой" лексический элемент. Этот пункт не затрагивает использование таких появлений.

16.5 Продукционное правило "ExtendedXMLChoiceValue"

16.5.1 "ExtendedXMLChoiceValue" :

ExtendedXMLChoiceValue::=
"<" & TagName AttributeList ">"
ExtendedXMLValue
"</" & TagName ">"
| ExtendedXMLValue

TagName::=
IdentifierOrModifiedIdentifier
IdentifierOrModifiedIdentifier::=
identifier
| QualifiedOrUnqualifiedName

16.5.2 "QualifiedOrUnqualifiedName" определено в 28.3.2. "QualifiedOrUnqualifiedName" должно использоваться, если имеется окончательная команда кодирования NAME (см. раздел 27), или окончательная команда кодирования NAMESPACE примененная к типу (см. раздел 28), иначе должен использоваться "идентификатор".

Примечание — Если "идентификатор" используется, то кодирование не может включать декларацию пространства имен XML по умолчанию с областью применения, которая включает использование этого идентификатора (см. 15.7).

16.5.3 "AttributeList" и его использование определены в 16.3 и в разделах, на которые 16.3 ссылается.

16.5.4 "ExtendedXMLValue" в обеих альтернативах для "ExtendedXMLChoiceValue" должно быть "ExtendedXMLValue" выбранной альтернативы типа выбора.

16.5.5 Вторая альтернатива "ExtendedXMLChoiceValue" должна использоваться если:

а) выбранная альтернатива для типа выбора имеет окончательную команду кодирования UNTAGGED (см. раздел 31);

б) тип выбора имеет окончательную команду кодирования USE-TYPE или USE-UNION (см. разделы 36 и 37).

Примечание — Это означает, что присутствие этих окончательных команд кодирования приводит к удалению тегов XML как определителей выбора, и определение выбора должно произойти другими средствами (см. разделы 36, 37).

16.6 Продукционные правила "ExtendedXMLSequenceValue" и "ExtendedXMLSetValue"

16.6.1 "ExtendedXMLSequenceValue" и "ExtendedXMLSetValue" — это:

ExtendedXMLSequenceValue::=
ExtendedXMLComponentValueList
| empty
ExtendedXMLSetValue::=
ExtendedXMLComponentValueList
| empty
ExtendedXMLComponentValueList::=
ExtendedXMLNamedValue
| ExtendedXMLComponentValueList ExtendedXMLNamedValue
ExtendedXMLNamedValue::=
"<" & TagName AttributeList ">"
ExtendedXMLValue
"</" & TagName ">"
| ExtendedXMLValue

16.6.2 "Пустые" альтернативы "ExtendedXMLSequenceValue" и "ExtendedXMLSetValue" должны использоваться, если никакой компонент типа последовательности или набора (до любой глубины) после разрешения всех ссылок на тип и после применения всех окончательных команд кодирования, не продуцирует "ExtendedXMLNamedValue".

Примечание — Это включает (но не ограничивает) случаи, в которых все компоненты помечены: DEFAULT или OPTIONAL, и все значения опущены; имеет окончательную команду кодирования UNTAGGED, и их значения имеют пустое кодирование; имеет окончательную команду кодирования ATTRIBUTE. Это также включает комбинации вышеупомянутого, и случая, в котором нотацией типа является SEQUENCE {} или SET {}.

16.6.3 "TagName" определено в 16.5.1. "QualifiedOrUnqualifiedName" в форме "IdentifierOrModifiedIdentifier" "TagName" должно использоваться, если и только если есть окончательная команда кодирования NAME или NAMESPACE, примененная к типу (см. раздел 28), иначе должен использоваться "идентификатор".

16.6.4 "AttributeList" и его использование определены в 16.3 и пунктах, на которые он ссылается.

16.6.5 "ExtendedXMLValue" в обеих альтернативах для "ExtendedXMLNamedValue" должно быть "ExtendedXMLValue" компонента типа последовательности или набора.

16.6.6 Вторая альтернатива для "ExtendedXMLSequenceValue" и "ExtendedXMLSetValue" должна использоваться, если и только если альтернатива имеет окончательную команду кодирования UNTAGGED (см. раздел 31).

16.7 Продукционные правила "ExtendedXMLSequenceOfValue" и "ExtendedXMLSetOfValue"

16.7.1 "ExtendedXMLSequenceOfValue" и "ExtendedXMLSetOfValue":

```
ExtendedXMLSequenceOfValue::=
        ExtendedXMLValueList
        | ExtendedXMLDelimitedItemList
        | empty
        | ExtendedXMLListValue
ExtendedXMLSetOfValue::=
        ExtendedXMLValueList
        | ExtendedXMLDelimitedItemList
        | empty
        | ExtendedXMLListValue
ExtendedXMLValueList::=
        ExtendedXMLValueOrEmpty
        | ExtendedXMLValueOrEmpty ExtendedXMLValueList
ExtendedXMLValueOrEmpty::=
        ExtendedXMLValue
        | "<" & TypeNameOrModifiedTypeName "/>"
ExtendedXMLDelimitedItemList::=
        ExtendedXMLDelimitedItem
        | ExtendedXMLDelimitedItem ExtendedXMLDelimitedItemList
ExtendedXMLDelimitedItem::=
        "<" & TypeNameOrModifiedTypeName AttributeList ">"
        ExtendedXMLValue
        "</" & TypeNameOrModifiedTypeName ">"
        |"<" & IdentifierOrModifiedIdentifier AttributeList ">"
        ExtendedXMLValue
        "</" & IdentifierOrModifiedIdentifier ">"
        | ExtendedXMLValue
```

16.7.2 Использование альтернатив "ExtendedXMLSequenceOfValue", "ExtendedXMLSetOfValue" и "ExtendedXMLValueList" должно быть в соответствии с использованием альтернатив "XMLSequenceOfValue", "XMLSetOfValue" и "XMLValueList" (соответственно), как определено в ИСО/МЭК 8824-1, 25 и 27, исключение составляет случай, когда команда кодирования GLOBAL-DEFAULTS с ключевым словом MODIFIED-ENCODINGS присутствует, тогда "ExtendedXMLValueList" не должен использоваться (см. также 9.2.7 ж).

16.7.3 "ExtendedXMLListValue" определено в 26.3.2. Эти альтернативы "ExtendedXMLSequenceOfValue" и "ExtendedXMLSetOfValue" должны использоваться, только если есть окончательная команда кодирования LIST (см. раздел 26), примененная к типу последовательность-из или набор-из.

16.7.4 Первая альтернатива "ExtendedXMLDelimitedItem" должна использоваться, если и только если тип последовательность-из или набор-из не содержит "идентификатор", и компонент не имеет окончательной команды кодирования UNTAGGED.

16.7.4.1 Если компонент типа последовательность-из или набор-из есть "typereference" или "ExternalTypeReference" (возможно с одним или более "TypePrefix"), то "TypeNameOrModifiedTypeName" должен быть "typereference" или "typereference" в "ExternalTypeReference", соответственно, возможно измененный в соответствии с любыми окончательными командами кодирования NAME и NAMESPACE, примененными к компоненту (см. раздел 27).

16.7.4.2 Если компонент типа последовательность-из или набор-из (после игнорирования любых появлений "TypePrefix") не "typereference" или "ExternalTypeReference", то "TypeNameOrModifiedTypeName" должно быть "xmlasn1typename", определенное в ИСО/МЭК 8824-1, таблица 4, соответствующее встроенному типу компонента, возможно измененного в соответствии с любой окончательной командой кодирования NAMESPACE, примененной к компоненту (см. раздел 28).

16.7.5 Вторая альтернатива "ExtendedXMLDelimitedItem" должна использоваться, если и только если тип последовательность-из или набор-из содержит "идентификатор", и компонент не имеет окончательной команды кодирования UNTAGGED. "IdentifierOrModifiedIdentifier" должен быть тем "идентификатором",

возможно измененным в соответствии с любыми окончательными командами кодирования NAME и NAMESPACE, примененными к компоненту (см. разделы 27 и 28).

16.7.6 Третья альтернатива "ExtendedXMLDelimitedItem" должна использоваться, если и только если компонент типа последовательность-из или набор-из имеет окончательную команду кодирования UNTAGGED (см. раздел 31).

16.7.7 "ExtendedXMLValue" во всех альтернативах для "ExtendedXMLDelimitedItem" должен быть "ExtendedXMLValue" повторного компонента типа последовательность-из или набор-из.

16.7.8 "TypeNameOrModifiedTypeName" в "ExtendedXMLValueOrEmpty" должен быть "xmlasn1typename", определенным в ИСО/МЭК 8824-1, таблица 4, соответствующий встроенному типу компонента, возможно измененного в соответствии с любой окончательной командой кодирования NAMESPACE, примененной к компоненту (см. раздел 28).

16.8 Продукционное правило "ModifiedXMLIntegerValue"

16.8.1 "ModifiedXMLIntegerValue":

```
ModifiedXMLIntegerValue::=
        ModifiedXMLSignedNumber
        | TextInteger
ModifiedXMLSignedNumber::=
        modifiedXMLNumber
        | "-" & modifiedXMLNumber
        | "+" & modifiedXMLNumber
```

16.8.2 Эта альтернатива "ExtendedXMLIntegerValue" (см. 16.4) должна использоваться, только если назначена команда кодирования GLOBAL-DEFAULTS с ключевым словом MODIFIED-ENCODINGS.

16.8.3 Лексическая единица "ModifiedXMLNumber" должна состоять из одной или более цифр.

Примечание 1 — Лексическая единица "ModifiedXMLNumber" отображена в целочисленное значение, она интерпретируется как десятичная нотация.

Примечание 2— Эта лексическая единица отличается от лексической единицы "число" (см. ИСО/МЭК 8824-1, 11.8) только тем, что разрешает любое число начальных (незначащих) нулей.

16.8.4 Любое положительное целое значение может быть закодировано с использованием либо первой, либо третьей альтернативы "ModifiedXMLSignedNumber" как опция кодера. Отрицательное целое значение должно быть закодировано с использованием второй альтернативы. Целое нулевое значения может быть закодировано с использованием любой из этих трех альтернатив как опция кодера.

16.8.5 "TextInteger" определено в ИСО/МЭК 8824-1, 18.9, и обеспечивает альтернативное кодирование (как опция кодера) для целочисленных значений, которые имеют определение "NamedNumber".

16.9 Продукционное правило "ModifiedXMLRealValue"

16.9.1 "ModifiedXMLRealValue":

```
ModifiedXMLRealValue::=
        ModifiedXMLNumericRealValue
        | XMLSpecialRealValue
        | XMLDecimalMinusZeroRealValue
ModifiedXMLNumericRealValue::=
        modifiedXMLRealNumber
        | "-" & modifiedXMLRealNumber
        | "+" & modifiedXMLRealNumber
```

16.9.2 Эта альтернатива "ExtendedXMLRealValue" (см. 16.4) должна использоваться, только если будет назначена команда кодирования GLOBAL-DEFAULTS с ключевым словом MODIFIED-ENCODINGS.

16.9.3 Лексическая единица "ModifiedXMLRealNumber" должна состоять из целой части, которая является рядом одной или более цифр, и опционально десятичной точки (.). Десятичная точка может опционально сопровождаться дробной частью, которая является одной или более цифрами. Целая часть, десятичная точка или дробная часть (какая бы ни была последним представлением) могут опционально сопровождаться "e" или "E" с показателем степени, состоящим из одной или более цифр и имеющим знак ("+" или "–" — опционально).

Примечание — Эта лексическая единица отличается от лексической единицы "действительное число" ("realnumber") (см. ИСО/МЭК 8824-1, подраздел 11.9) только потому, что разрешает любое число начальных нулей в показателе степени.

16.9.4 Любое положительное действительное значение и действительное значение плюс нуль могут быть закодированы с использованием первой или третьей альтернативы "ModifiedXMLNumericRealValue" как опция кодера. Любое отрицательное действительное значение должно быть закодировано с использованием второй альтернативы "ModifiedXMLNumericRealValue". Действительное значение минус нуль должно быть закодировано с использованием второй альтернативы.

16.9.5 "XMLDecimalMinusZeroRealValue" определено в 21.3.2 и должно использоваться только, как определено в том пункте.

Примечание — Команда кодирования DECIMAL, определенная в 21.3.2, обеспечивает это продукционное правило как альтернативное представление для положительного нулевого абстрактного действительного значения, но требует, чтобы отрицательное нулевое абстрактное значение было исключено из типа, к которому оно применено.

17 Команда кодирования ANY-ATRIBUTE

17.1 Общие положения

17.1.1 "AnyAttributesInstruction":

```
AnyAttributesInstruction::=
        ANY-ATTRIBUTES
        TargetList
        NamespaceRestriction ?
NamespaceRestriction::=
        FROM URIList
        | EXCEPT URIList
URIList::=
        QuotedURIorAbsent
        | URIList QuotedURIorAbsent
QuotedURIorAbsent::=
        QuotedURI
        | ABSENT
```

17.1.2 Продукционное правило "TargetList" определено в 13.2.

17.1.3 "QuotedURI" определено в 28.1.1.

17.1.4 Данная команда кодирования назначается типу ACH.1, который является типом последовательность-из или набор-из с компонентом UTF8String, значение которого обеспечивает нуль, один или более имен атрибута и значений (по одному в каждом UTF8String), каждое из которых подчиняется любому присутствующему ограничению пространства имен "NamespaceRestriction".

17.1.5 Контент каждой UTF8String закодирован как атрибут "Attribute" включающего XML элемента. Имя компонента последовательность-из или набора-из игнорируется.

17.1.6 Разделы FROM и EXCEPT (если присутствуют) идентифицируют перечни имен пространства имен или специальное ключевое слово ABSENT.

17.1.7 FROM ограничивает имена атрибутов таким образом, что они должны быть именами пространства имен ограниченного применения из одного из специфицированных пространств имен. Если ABSENT присутствует в "URIList", неуточненные имена также могут использоваться.

17.1.8 EXCEPT позволяет исключать перечисленные имена пространства имен ограниченного применения из любого пространства имен. Эта команда допускает также неуточненные имена, если только ABSENT не присутствует в "URIList".

17.2 Ограничения

17.2.1 Тип ACH.1 не должен иметь эту окончательную команду кодирования, если она не типа последовательность-из или набор-из с компонентом типа UTF8String.

17.2.2 Тип с этой окончательной командой кодирования должен использоваться только как компонент типа включающей последовательности или набора, и компонент не должен быть маркирован как OPTIONAL или DEFAULT. Должен быть только один такой компонент в включающем типе.

17.2.3 Тип последовательность-из или набор-из с этой окончательной командой кодирования обязательно должен иметь ограничение, применяемое к ней, которое задает формат и контент, определенные в 18.2.6—18.2.11 при каждом появлении UTF8String или определенные другим способом.

Примечание — Рекомендуется, чтобы ограничение типов набор-из или последовательность-из выражалось как:

(CONSTRAINED BY

{/* Каждая UTF8String должна соответствовать "AnyAttributeFormat", определенному в ИСО/МЭК 8825-4, 18. */})

17.2.4 Не должно быть окончательной команды кодирования UNTAGGED, применяемой к типу, который имеет эту окончательную команду кодирования или к включающему типу.

17.2.5 Каждый "URIList" должен содержать не более одного появления ABSENT и не должен содержать двух идентичных "QuotedURI".

17.2.6 Формат каждой UTF8String должен соответствовать продукционному правилу "AnyAttributeFormat":

AnyAttributeFormat::=

URI ?

NCName & "=" & xmlcstring

17.2.7 Для определения продукционного правила "URI" см. 28.1.4, для определения продукционного правила "NCName" см. 28.1.7. Лексическая единица "xmlcstring" определена в ИСО/МЭК 8824-1, 11.15.

17.2.8 Если имеется "NamespaceRestriction" команды кодирования FROM, то "URI" в "AnyAttributeFormat" должен быть "URI" в "QuotedURI" в "URIList" и может отсутствовать, только если ключевое слово a ABSENT встречается в "URIList".

17.2.9 Если имеется "NamespaceRestriction" команды кодирования EXCEPT, то "URI" в "AnyAttributeFormat" не должен быть "URI" в "QuotedURI" в "URIList" и не может отсутствовать, если ключевое слово ABSENT встречается в "URIList".

17.2.10 "Xmlcstring" должно быть синтаксически правильным значением XML атрибута (определенным в W3C XML, раздел 3), которому предшествует и после которого следует или единственный символ APOSTROPHE (см. раздел 38), или единственный символ QUOTATION MARK (см. раздел 34).

17.2.11 Применение этой команды кодирования и команды кодирования ATTRIBUTE к различным компонентам включающего типа не будет нарушать условий, приведенных в 19.3.11.

17.2.12 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

17.2.13 Тип с этой окончательной командой кодирования не должен иметь ни одной из перечисленных окончательных команд кодирования: LIST, PI-OR-COMMENT или UNTAGGED.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не могут появляться вместе с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ELEMENT, ATTRIBUTE, BASE64, DECIMAL, DEFAULT-FOR-EMPTY, EMBED-VALUES, TEXT, USE-NIL, USE-NUMBER, USE-ORDER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE.

17.2.14 В "TargetList" не должно быть никакой уточняющей информации.

17.3 Действие на результат кодирования

17.3.1 Если тип кодируется как тип верхнего уровня, эта команда кодирования должна игнорироваться.

17.3.2 "ExtendedXMLNamedValue" для этого компонента не должна включаться в "ExtendedXMLSequenceValue" или "ExtendedXMLSetValue" типа включающей последовательности или набора. Вместо этого значение включающего типа должно кодироваться с использованием значения каждой UTF8String как атрибута "Attribute" (см. 19) включающего элемента, специфицированного ниже.

17.3.3 Кодер должен:

а) обрабатывать каждый "URI", который присутствует в UTF8String, как требующий, чтобы следующий "NCName" (имя атрибута) был пространством имен ограниченного применения с пространством имен, определенным "URI", и рассматривать отсутствие "URI" в UTF8String как признак того, что следующий "NCName" не должен быть пространством имен ограниченного применения и должен тогда удалить "URI" из UTF8String;

б) вставить в результат кодирования любые необходимые декларации пространства имен с масштабами охвата, которые включают вставляемые атрибуты для обеспечения того, чтобы необходимые уточнения пространства имен "NCName", идентифицированные в (а), были достигнуты;

в) вставить каждый UTF8String (после удаления "URI") как атрибут в включающем элементе, включая префиксы пространства имен как обязательные перед каждым "NCName" для обеспечения того, чтобы требования, приведенные в (а) были удовлетворены.

17.3.4 Порядок всех атрибутов в включающем элементе (определяемый присутствием одного или более компонента включающего типа с окончательной командой кодирования ATTRIBUTE или ANY-ATTRIBUTES) есть опция кодера.

17.3.5 Декодер EXTENDED-XER должен генерировать UTF8String в формате ограничений (см.18.2.6) для каждого атрибута в включающем элементе, который не входит в управляющее пространство имен и имя которого не совпадает с именем идентификатора (возможно, модифицированного в соответствии с любой из окончательных команд кодирования NAME или NAMESPACE encoding) другого компонента включающего типа с окончательной командой кодирования ATTRIBUTE.

18 Команда кодирования ANY-ELEMENT

18.1 Общие положения

18.1.1 "AnyElementInstruction":

AnyElementInstruction::=
    ANY-ELEMENT
    TargetList
    NamespaceRestriction ?

18.1.2 Продукционное правило "TargetList" определено в 13.2.

18.1.3 "NamespaceRestriction" определено в 17.1.

18.1.4 Эта команда кодирования дает типу ACH.1 возможность для UTF8String обеспечивать спецификацию единственного XML элемента.

Примечание — Контент и атрибуты XML элемента не ограничиваются. Он может иметь атрибуты или дочерние элементы, и имена этих дочерних элементов могут быть составными или несоставными и не быть подверженными влиянию "NamespaceRestriction".

18.1.5 Если присутствует ограничение пространства имен "NamespaceRestriction", то требуется имя элемента для удовлетворения "NamespaceRestriction" (см. 17.1.6 — 17.1.8), иначе пространство имен не имеет ограничения.

18.1.6 UTF8String с этой окончательной командой кодирования может быть корневого типа кодирования или может быть компонентом типа выбора, последовательности, набора, последовательности-из или набора-из. Если это тип высшего уровня, имя ссылки на тип игнорируется. Если это — компонент, имя компонента игнорируется.

18.2 Ограничения

18.2.1 Тип ACH.1 не должен иметь этой окончательной команды кодирования, если только это не тип UTF8String. Требуется компонент, чтобы иметь ограничение, примененное к нему, что подразумевает формат и контент, определенные в 18.2.4—18.2.9 ссылкой на раздел 18 или другие.

Примечание — Рекомендуется ограничение UTF8String выражать как:
(CONSTRAINED BY
{/* Должна соответствовать "AnyElementFormat" определенному в ИСО/МЭК 8825-4, раздел 19. */})

18.2.2 Не должно быть окончательной команды кодирования UNTAGGED, применяемой к типу.

18.2.3 Каждый "URIList" должен содержать не более одного появления ABSENT и не содержать двух идентичных "QuotedURI".

18.2.4 Формат абстрактных значений UTF8String должен соответствовать продукционному правилу "AnyElementFormat":

AnyElementFormat::=
    xmlcstring

18.2.5 "Xmlcstring" должна быть синтаксически правильным XML элементом, определенным в W3C XML 1.0 и W3C XML Namespaces.

18.2.6 "Xmlcstring" должна использовать только префиксы пространства имен, которые объявлены в декларациях пространства имен, присутствующих в "xmlcstring". Если существуют составные имена, не снабженные префиксами, должна присутствовать соответствующая декларация пространства имен по умолчанию.

18.2.7 Значение UTF8String не должно нарушать условия пункта 9.2.11.

18.2.8 Если FROM имеет "NamespaceRestriction", то (самое удаленное) имя элемента в "AnyElementFormat" должно быть "URI" в "QuotedURI" в "URIList", и может отсутствовать, только если ключевое слово ABSENT появляется в "URIList".

18.2.9 Если EXCEPT имеет "NamespaceRestriction", то (самое удаленное) имя элемента в "AnyElementFormat" не должно быть "URI" в "QuotedURI" в "URIList", и не должно отсутствовать, если ключевое слово ABSENT появляется в "URIList".

18.2.10 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

18.2.11 Тип с этой окончательной командой кодирования не должен также иметь никакой из перечисленных окончательных команд кодирования: ATTRIBUTE, BASE64, DEFAULT-FOR-EMPTY, PI-OR-COMMENT, UNTAGGED или WHITESPACE.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не могут появляться вместе с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, DECIMAL, EMBED-VALUES, LIST, TEXT, USE-NIL, USE-NUMBER, USEORDER, USE-QNAME, USE-TYPE, USE-UNION.

18.2.12 В "TargetList" не должно быть никакой уточняющей информации.

18.3 Действие на результат кодирования

18.3.1 EXTENDED-XER кодер должен включать абстрактное значение UTF8String при кодировании, как XML элемент вместо XML элемента, который бы иначе был сгенерирован для этого компонента (при игнорировании идентификатора компонента) или для корневого типа. Включенный элемент должен быть идентичен абстрактному значению UTF8String, за исключением ситуации, специфицированной в 18.3.2.

18.3.2 Любые декларации пространства имен, которые присутствуют в первом стартовом теге (или теге пустого элемента) элемента и идентичны декларациям пространства имен, появляющимся в точке вставки, могут быть (не обязательно) удалены в качестве опции кодера.

Примечание — Изменение, перемещение или удаление других деклараций пространства имен в UTF8String не разрешены, поскольку такие действия могут влиять на пространство имен и уточнение XML Q-имен, присутствующих в контенте элемента или значениях атрибута и в общем случае для кодера невозможно определить, являются ли такие значения контента или атрибута Q-именами.

18.3.3 EXTENDED-XER декодер должен сгенерировать формат (см.18.2.4) из входящего XML документа как абстрактное значение UTF8String.

18.3.4 Декодер должен включать в первом стартовом теге (или теге пустого элемента) в абстрактном значении UTF8String атрибуты декларации пространства имен для всех деклараций пространства имен, которые принадлежат одному множеству с декодируемым элементом, но которые не присутствуют в стартовом теге этого элемента.

19 Команда кодирования ATTRIBUTE

19.1 Общие положения

19.1.1 "AttributeInstruction":

AttributeInstruction::=
    ATTRIBUTE
    TargetList

19.1.2 Продукционное правило "TargetList" определено в 13.2.

19.1.3 Эта команда кодирования определяет, что тип АСН.1, который может кодироваться посимвольно, следует кодировать как XML атрибут.

Примечание — Частный (но важный) пример типа, который может кодироваться посимвольно, есть тип выбора (все, альтернативы являются типами, которые могут кодироваться посимвольно), имеющий окончательную команду кодирования USE-UNION.

19.2 Ограничения

19.2.1 Тип АСН.1 не должен иметь этой окончательной команды кодирования, если он не имеет, по крайней мере, одного кодирования "ExtendedXMLValue" (принимая во внимание опции кодера) для каждого из его абстрактных значений, которые не содержат тегов XML и не основаны на использовании "xmlhstring" (если тип является открытым типом или типом строки октетов) или "xmlbstring" (если тип является типом строки битов), или на окончательных командах кодирования UNTAGGED, ATTRIBUTE, или ANY-ATTRIBUTE, примененных к его компонентам (если тип является типом последовательности или набора) чтобы достичь этого.

Примечание 1 — Подразумевается, что тип строки с ограниченным набором символов с окончательной командой кодирования ATTRIBUTE должен иметь такое ограничение, чтобы не содержать никаких управляющих символов, перечисленных в ИСО/МЭК 8824-1, таблица 3 (последовательности управляющих символов в "xmlcstring"), или должен иметь окончательную команду кодирования BASE64.

Примечание 2 — Не должны включаться открытые типы или типы строки октетов и строки битов с CONTAINING без ENCODED BY, так как их "ExtendedXMLValue" может содержать теги, если только они не кодируются как "xmlhstring".

Примечание 3 — Признано, что некоторые инструменты АСН.1 не могут обеспечить статистический контроль за тем, что вышеуказанное ограничение удовлетворяется для всех абстрактных значений, а кодеры, соответствующие правилам кодирования Extended XER, не могут генерировать результаты кодирования, в которых "ExtendedXMLValue" нарушает это ограничение (см. 19.3.14).

19.2.2 Тип с этой окончательной командой кодирования будет использоваться только как компонент типа последовательности или набора.

Примечание — Компонент может быть OPTIONAL или DEFAULT.

19.2.3 Не должно быть окончательной команды кодирования UNTAGGED для типа, который имеет эту окончательную команду кодирования или для включающего типа, который содержит ее как компонент.

19.2.4 Если окончательные команды кодирования для других компонентов включающего типа включают или эту команду кодирования, или команду кодирования ANY-ATTRIBUTES, условие, приведенное в 19.3.11, не будет нарушаться.

19.2.5 Тип с этой окончательной командой кодирования не должен также иметь любой из окончательных команд кодирования: ANY-ELEMENT, DEFAULT-FOR-EMPTY, PI-OR-COMMENT или UNTAGGED.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не могут использоваться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, EMBED-VALUES, USE-NIL, USE-ORDER, USE-TYPE.

19.2.6 Не должно быть уточняющей информации в "TargetList".

19.3 Действие на результат кодирования

19.3.1 Если тип кодируется как тип высшего уровня, то эта команда кодирования должна быть проигнорирована.

19.3.2 "ExtendedXMLNamedValue" этого компонента не должно быть включено в "ExtendedXMLSequenceValue" или "ExtendedXMLSetValue" типа включающей последовательности или набора. Вместо этого значение компонента (если присутствует) должно кодироваться как "Attribute" (см. 19.3.3—19.3.15) включающего элемента.

19.3.3 Продукционное правило "Attribute":

```
Attribute::=
        AttributeName
        "="
        QuotedValue
AttributeName::=
        IdentifierOrModifiedIdentifier
        | ControlAttributeName
QuotedValue::=
        DoubleQuotedValue
        | SingleQuotedValue
DoubleQuotedValue::=
        """ & CharacterEncodableValue & """
SingleQuotedValue::=
        """ & CharacterEncodableValue & """
ControlAttributeName::= QualifiedName
CharacterEncodableValue::= ExtendedXMLValue
```

19.3.4 Продукционное правило "IdentifierOrModifiedIdentifier" определено в 16.5.1, и его использование в контексте этой команды кодирования определено в 16.6.3.

19.3.5 Продукционное правило "ControlAttributeName" не используется непосредственно этим разделом. Все "QualifiedName"s в этом продукционном правиле взяты из управляющего пространства имен (см. 15.9). Такие атрибуты генерируются только в соответствии с разделами 32 и 36, а непредвиденные управляющие атрибуты необходимо акцептовать декодерам (см. 9.2.10).

19.3.6 "QualifiedName" определено в 28.3.2.

19.3.7 "ExtendedXMLValue" определено в 16.4.

19.3.8 "AttributeName" должно быть или "идентификатором" компонента, который имеет эту окончательную команду кодирования, или, если есть окончательные команды кодирования NAME или NAMESPACE, "QualifiedOrUnqualifiedName" определяется этими командами кодирования, как специфицировано в разделах 27 и 28.

19.3.9 "CharacterEncodableValue" в "QuotedValue" атрибута (см. 19.3.3) должно быть "ExtendedXMLValue" этого типа, возможно, модифицированным, как определено в 19.3.12–19.3.15.

19.3.10 Порядок, в котором атрибуты появляются в "AttributeList", является опцией кодировщика: будут ли они генерироваться этой командой кодирования или командой кодирования ANY-ATTRIBUTES.

Примечание — Никакой семантики нельзя накладывать на порядок атрибутов в любой команде EXTENDED-XER. Это ограничение является требованием W3C XML 1.0, 3.1.

19.3.11 Когда "AttributeList" в варианте кодирования содержит множественные атрибуты, то для любых двух атрибутов в перечне:

а) если имена "AttributeName"s двух атрибутов — оба неуточненные имена, то они должны быть различными;

б) если имена"AttributeName"s двух атрибутов — оба имени уточненных пространств имен, то либо они должны иметь различные пространства имен, либо они должны быть различными именами одного и того же пространства имен.

Если это условие будет нарушаться применением окончательных команд кодирования для любого абстрактного значения типа высшего уровня, который кодируется, такое использование команд кодирования будет неправомочным.

19.3.12 Если "QuotedValue" является "DoubleQuotedValue", и "ExtendedXMLValue" в "CharacterEncodableValue" содержит символ КАВЫЧКИ (QUOTATION MARK (34)) то этот символ должен быть заменен символами:

"

или, как опция кодировщика, управляющей последовательностью формы &#n; или &#xn;, определенными ИСО/МЭК 8824-1, 11.15.8.

19.3.13 Если "QuotedValue" является "SingleQuotedValue" и "ExtendedXMLValue" в "CharacterEncodableValue" содержит символ АПОСТРОФ (APOSTROPHE (39)), то этот символ должен быть заменен символами:

'

или, как опция кодировщика, управляющей последовательностью формы &#n; или &#xn;, определенными ИСО/МЭК 8824-1, 11.15.8.

19.3.14 "ExtendedXMLValue" в "CharacterEncodableValue" должно быть одним из результатов кодирования типа, который можно кодировать посимвольно и который не содержит тегов XML.

19.3.15 Если "ExtendedXMLValue" содержит символы ГОРИЗОНТАЛЬНАЯ ТАБУЛЯЦИЯ (HORIZONTAL TABULATION (9)), ПЕРЕВОД СТРОКИ (LINE FEED (10)), или ПЕРЕВОД КАРЕТКИ (CARRIAGE RETURN (13)), то эти символы должны быть заменены в "ExtendedXMLValue" управляющими последовательностями формы "&#n;" или "&#xn;", определенными в ISO/IEC 8824-1, 11.15.8.

20 Команда кодирования BASE64

20.1 Общие положения

20.1.1 "Base64Instruction":

Base64Instruction:: =

BASE64

TargetList

20.1.2 Продукционное правило "TargetList" определено в 13.2.

20.1.3 Эта команда кодирования может быть назначена типу строка октетов, "открытый" или любому типу "ограниченная строка символов".

20.1.4 Применение этой окончательной команды кодирования к типу строки октетов или открытому типу удаляет опцию шестнадцатеричного кодирования, но позволяет опцию кодирования Base64 (как определено в запросе на комментарии IETF 2045, 6.8). Применение этой окончательной команды кодирования к типу ограниченная строка символов требует, чтобы значение типа ограниченной строки символов было закодировано как Base64.

20.2 Ограничения

20.2.1 Если окончательная команда кодирования для типа ACH.1 будет содержать команду кодирования BASE64 , то тип должен быть одним из:

а) строка октетов;

б) открытый;

в) ограниченная строка символов.

20.2.2 Тип с этой окончательной командой кодирования не должен иметь окончательных команд кодирования ANY-ELEMENT или WHITESPACE.

Примечание — Нижеперечисленные окончательные команды кодирования не могут появляться вместе с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, DECIMAL, EMBED-VALUES, LIST, TEXT, USE-NIL, USE-NUMBER, USEORDER, USE-QNAME, USE-TYPE, USE-UNION.

20.2.3 В "TargetList" не должно быть уточняющей информации.

20.3 Действие на результат кодирования

20.3.1 Эта команда кодирования затрагивает только "ExtendedXMLValue" типа, к которому она применяется. Она требует использования первой или второй альтернативы для "ExtendedXMLOctetStringValue" и "ExtendedXMLOpenTypeFieldVal" (как опция кодера), запрещая третью альтернативу (см. 16.4). Она требует использования второй альтернативы для "ExtendedXMLRestrictedCharacterStringValue" (см. 16.4).

20.3.2 "Base64XMLOctetStringValue":

Base64XMLOctetStringValue:: =

XMLBase64String

"XMLBase64String" определен в 20.3.6.

20.3.3 ИСО/МЭК 8824-1, пункт 22.4, применяется.

20.3.4 "Base64XMLOpenTypeFieldVal":

Base64XMLOpenTypeFieldVal:: =

XMLBase64String

20.3.5 "Base64XMLRestrictedCharacterStringValue":

Base64XMLRestrictedCharacterStringValue:: =

XMLBase64String

20.3.6 "XMLBase64String":

XMLBase64String:: =

XMLRestrictedCharacterStringValue

"XMLRestrictedCharacterStringValue" должно быть результатом кодирования контента при передаче Content-Transfer-Encoding, определенное в запросе на комментарии IETF 2045, пункт 6.8, за исключением того, что предел в 76 символов не применяется, и "пробел с переходом" (см. 7.1.5), разрешен в любой позиции результата кодирования "XMLBase64String".

Примечание — IETF RFC, 2045 определяет присутствие концов строки, делящих результат кодирования на строки максимум 76 символов, но это не требуется при EXTENDED-XER кодированиии. Это также позволяет вставить пробел "white-space" в любую позицию результата кодирования Base64.

20.3.7 Если команда кодирования Base64 применяется к типу "ограниченная строка символов", то каждый символ в строке символов должен быть закодирован с UTF-8 (см. ИСО 10646, приложение D). Получающиеся октеты для всей строки символов должны быть закодированы в символы, как определено в IETF RFC 2045, пункт 6.8, и получающиеся символы должны сформировать "ExtendedXMLValue".

21 Команда кодирования DECIMAL

21.1 Общие положения

21.1.1 "DecimalInstruction":

DecimalInstruction:: =

DECIMAL

TargetList

21.1.2 Продукционное правило "TargetList" определено в 13.2.

21.1.3 Назначение этой команды кодирования состоит в изменении кодирования действительного типа так, чтобы запрещалась экспоненциальная нотация, и дефис, сопровождаемый "0", обозначал значение плюс нуль вместо значения минус нуль.

Примечание — Значение минус нуль не может быть представлено.

21.2 Ограничения

21.2.1 Эта команда кодирования используется только для действительного типа.

21.2.2 Действительный тип, к которому применена эта команда кодирования, должен быть ограничен таким способом, что значения минус нуль, MINUS-INFINITY, PLUS-INFINITY и NOT-A-NUMBER не разрешаются, и base есть 10.

Примечание — Рекомендуются следующие ограничения:
((WITH COMPONENTS {..., base(10)})
(ALL EXCEPT (-0 | MINUS-INFINITY | PLUS-INFINITY | NOT-A-NUMBER))

21.2.3 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

21.2.4 Тип с этой окончательной командой кодирования может иметь любые другие окончательные команды кодирования, разрешенные для этого типа.

Примечание — Нижеперечисленные окончательные команды кодирования не могут появляться вместе с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, ANY-ELEMENT, BASE64, EMBED-VALUES, LIST, TEXT, USE-NIL, USE-NUMBER, USE-ORDER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE.

21.2.5 В "TargetList" не должно быть уточняющей информации.

21.3 Действие на результат кодирования

21.3.1 "ModifiedXMLRealNumber" (см. 16.9.3) не должно содержать "e" или "E", сопровождаемое показателем степени.

Примечание — Все абстрактные значения, включая те, которые являются очень большими или очень малыми реальными числами, должны быть закодированы как целая часть, опционально сопровождаемая десятичным разделом и дробной частью.

21.3.2 Действительное значение плюс нуль может быть закодировано в качестве опции кодера как "XMLDecimalMinusZeroRealValue", определенное следующим образом:

XMLDecimalMinusZeroRealValue:: =
"-" & modifiedXMLRealNumber

где "ModifiedXMLRealNumber" ограничено условием 21.3.1 и не содержит никаких цифр кроме нуля.

Примечание — Вышеупомянутое не может быть спутано с действительным значением минус нуль, потому что значение минус нуль удалено обязательным ограничением, которое применяется к действительному типу (см. 21.2.2).

22 Команда кодирования DEFAULT-FOR-EMPTY

22.1 Общие положения

22.1.1 "DefaultForEmptyInstruction":

DefaultForEmptyInstruction::=
DEFAULT-FOR-EMPTY
TargetList
AS Value

22.1.2 Продукционное правило "TargetList" определено в 13.2.

22.1.3 Эта команда кодирования определяет абстрактное значение, которое может быть закодировано при EXTENDED-XER кодировании (как выбор кодера) как "пустая" альтернатива для "ExtendedXMLValue" для типа (см. 16.4), который кодируется как единственный контент элемента XML.

Примечание — Этот механизм по умолчанию поддерживает присутствие элемента XML без контента (типично, но не обязательно, закодированный как тег пустого элемента). Это отличается от использования DEFAULT ACH.1, которая связана с отсутствием "ExtendedXMLNamedValue" компонента последовательности или набора.

22.1.4 "TargetList" не должен использовать ключевое слово ALL и должен определять единственную цель.

22.1.5 Пять различных случаев, когда эта команда кодирования может использоваться, представлены ниже.

22.1.5.1 Первый случай — команда назначается непосредственно посимвольно кодируемому типу, который не является нетегированным UNTAGGED (см. раздел 31). Если включающий элемент приложения имеет пустой контент, то этот пустой контент представляет специфицированное значение "Value" посимвольно кодируемого типа (который управляет значением "Value").

22.1.5.2 Второй случай — команда назначается (NOT UNTAGGED, NOT EMBED-VALUES и NOT USE-NIL) типу последовательность, содержащему UNTAGGED посимвольно кодируемый компонент, при кодировании которого формируется единственный контент (для всех абстрактных значений типа последовательность) включающего элемента типа последовательность. Если включающий элемент типа последовательность имеет пустой контент, то этот пустой контент представляет специфицированное значение "Value" посимвольно кодируемого типа (который управляет значением "Value").

Примечание — Посимвольно кодируемый компонент может быть единственным контентом, потому что это — единственный компонент, или он может быть единственным контентом, потому что все другие компоненты имеют окончательные команды кодирования ATTRIBUTE (см. раздел 19), или ANY-ATTRIBUTES (см. раздел 17).

22.1.5.3 Третий случай — команда назначается (NOT UNTAGGED, NOT EMBED-VALUES и NOT USE-NIL) типу последовательность с окончательной командой кодирования EMBED-VALUES (см. 24.3.1.4). Если включающий элемент типа последовательность имеет пустой контент, то этот пустой контент представляет абстрактное значение типа последовательности, который иначе продуцировал бы контент, исключительно определенный значением "Value" единственной UTF8String в последовательности-из EMBED-VALUES (UTF8String управляет значением "Value").

22.1.5.4 Четвертый случай — команда назначается (NOT UNTAGGED, NOT EMBED-VALUES) типу последовательность с окончательной командой кодирования USE-NIL (см. раздел 32), чей компонент OPTIONAL — посимвольно кодируемый тип. Если включающий элемент типа последовательность имеет атрибут идентификации пустого множества со значением true, DEFAULT-FOR-EMPTY не влияет на смысл результата кодирования. Если включающий элемент типа последовательность имеет атрибут идентификации пустого множества со значением false (или не имеет атрибута идентификации пустого множества) и имеет пустой контент, то этот пустой контент представляет специфицированное значение "Value" компонента OPTIONAL (чей тип управляет значением "Value").

22.1.5.5 Пятый случай — команда назначается (NOT UNTAGGED) типу последовательность с окончательной командой кодирования EMBED-VALUES (см. 24.3.1.4) и с окончательной командой кодирования USE-NIL (см. раздел 32), чей компонент OPTIONAL — типа последовательность. Если включающий элемент типа последовательность имеет атрибут идентификации пустого множества со значением true, DEFAULT-FOR-EMPTY не влияет на результат кодирования. Если включающий элемент типа последовательность имеет атрибут идентификации пустого множества со значением false (или не имеет атрибута идентификации пустого множества) и имеет пустой контент, то этот пустой контент представляет абстрактное значение типа последовательность, который иначе продуцировал бы контент, определенный значением "Value" единственной UTF8String в последовательности-из EMBED-VALUES (UTF8String управляет значением"Value").

22.1.6 "Value" определено ИСО/МЭК 8824-1, 16.7.

Примечание — Это позволяет использовать эталонное значение, определенное в модуле или импортированное в него (модуль). Эталонное значение может быть определено с использованием нотации значений XML (XML Value Notation), но такая нотация не может использоваться непосредственно в "DefaultForEmptyInstruction".

22.2 Ограничения

22.2.1 Если окончательные инструкции кодирования для типа ACH.1, который является NOT UNTAGGED посимвольно кодируемым типом, содержит команду кодирования DEFAULT-FOR-EMPTY, то этот тип не должен быть компонентом (ACH.1 SEQUENCE или SET) со значением DEFAULT ACH.1.

Примечание — Это ограничение не является строго обязательным, оно должно исключить путаницу между нормальным механизмом определения ACH.1 и механизмом по умолчанию EXTENDED-XER.

22.2.2 Эта команда кодирования должна быть назначена только:

а) посимвольно кодируемому типу без окончательной команды кодирования UNTAGGED;

б) типу последовательность NOT UNTAGGED без окончательных команд кодирования EMBED-VALUES или USE-NIL, одним из компонентов которых является посимвольно кодируемый тип с окончательной командой кодирования UNTAGGED и все другие компоненты (если есть хотя бы один) имеют окончательные команды кодирования ATTRIBUTE или ANY-ATTRIBUTES;

в) типу последовательность NOT UNTAGGED, без окончательной команды кодирования USE-NIL, но с окончательной командой кодирования EMBED-VALUES (см. 24.3.1.4);

г) типу последовательность NOT UNTAGGED без окончательной команды кодирования EMBED-VALUES, но с окончательной командой кодирования USE-NIL, OPTIONAL компонент которой — посимвольно кодируемый тип;

д) типу последовательность NOT UNTAGGED с окончательной командой кодирования EMBED-VALUES и с окончательной командой кодирования USE-NIL, OPTIONAL компонент которой — типа последовательность.

22.2.3 Если применяется ограничение 22.2.2 (а) и "empty" — допустимое значение "ExtendedXMLValue" для одного из абстрактных значений (скажем, V), возможно, ограниченного типа, и V отличается от значения "Value" в "DefaultForEmptyInstruction", то должен быть, по крайней мере, один альтернативный результат кодирования для V.

22.2.4 Если применяются ограничения 22.2.2 (б) или (г), и "empty" — допустимое значение "ExtendedXMLValue" для одного из абстрактных значений (скажем, V) UNTAGGED компонента (случай б) или OPTIONAL компонента, (случай г), и V отличается от значения "Value" в "DefaultForEmptyInstruction", то должен быть, по крайней мере, один альтернативный результат кодирования для V.

Примечание — Признается, что некоторые инструменты АСН.1 не в состоянии статически проверить, что вышеупомянутые ограничения соответствуют всем абстрактным значениям, но соответствующие кодировщики не могут продуцировать результаты кодирования, в которых "ExtendedXMLValue" нарушает это ограничение.

22.2.5 Если применяется ограничение 22.2.2 (в), то тип SEQUENCE должен быть ограничен так, чтобы (без DEFAULT-FOR-EMPTY) не было абстрактного значения, которое продуцировало бы пустой контент для заключающего элемента.

22.2.6 Если посимвольно кодируемый тип [случай 22.2.2 (а)] с этой окончательной командой кодирования имеет тип, являющийся типом последовательность-из или набор-из с окончательной командой кодирования LIST, или типам выбора с окончательной командой кодирования USE-UNION, тогда эта окончательная команда кодирования должна игнорироваться.

22.2.7 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

22.2.8 Тип с этой окончательной командой кодирования не должен иметь окончательных команд кодирования ANY-ELEMENT, ATTRIBUTE или UNTAGGED.

Примечание — Окончательные команды кодирования ANY-ATTRIBUTES, USE-TYPE не могут применяться с этой окончательной командой кодирования, их применение к этому типу запрещено.

22.2.9 В "TargetList" не должно быть уточняющей информации.

22.3 Действие на результат кодирования

22.3.1 Эта команда кодирования действует только на значение "ExtendedXMLValue" типа, который управляет "значением" "Value" (см. 22.1.5).

22.3.2 Результат кодирования "ExtendedXMLValue" абстрактного значения, специфицированного "Value" будет (как выбор кодера) одним из следующих:

а) результат кодирования "ExtendedXMLValue" того значения, которое было бы продуцировано, если бы DEFAULT-FOR-EMPTY отсутствовал (нормальный результат кодирования);

б) "пустой" "empty".

Примечание — От декодеров требуется, чтобы они принимали и нормальный результат кодирования и "пустой" как обозначение пустого по умолчанию значения DEFAULT-FOR-EMPTY.

22.3.3 Если применяется ограничение 22.2.2 (а) и "пустой" "empty" есть допустимое значение "ExtendedXMLValue" для одного из абстрактных значений (скажем, V) типа и V отличается от "значения" "Value", специфицированного в "DefaultForEmptyInstruction", тогда любая из альтернатив результата кодирования для V должна использоваться (как опция кодера) вместо "пустого" "empty".

22.3.4 Если применяются ограничения 22.2.2 (б) или (г) и "пустое" "empty" — допустимое значение "ExtendedXMLValue" для одного из абстрактных значений (скажем, V) UNTAGGED компонента [случай (б)] или OPTIONAL компонента [случай (г)] и V отличается от "значения" "Value", специфицированного в "DefaultForEmptyInstruction", тогда любой из альтернативных результатов кодирования для V должен использоваться (как опция кодера) вместо "пустого" "empty".

22.3.5 Если применяется ограничение 22.2.2 (в), действие этой команды кодирования определено в 24.3.1.4 и 24.3.1.5.

22.3.6 Если применяется ограничение 22.2.2 (д), действие этой команды кодирования определено в 24.3.1.6.

23 Команда кодирования ELEMENT

23.1 Общие положения

23.1.1 "ElementInstruction":

ElementInstruction ::=

ELEMENT

TargetList

23.1.2 Продукционное правило "TargetList" определено в 13.2.

23.1.3 Эта команда кодирования синонимична с NOT UNTAGGED и не подразумевает никакой семантики кроме NOT UNTAGGED.

23.2 Ограничения

23.2.1 В "TargetList" не должно быть уточняющей информации.

23.2.2 Эта команда кодирования не должна использоваться как предварительно установленная команда кодирования, снабженная префиксом, в комбинации ни с одной из команд, снабженных префиксами ANY-ATTRIBUTES, ANY-ELEMENT или ATTRIBUTE, чтобы избежать разночтения.

23.3 Действие на результат кодирования

Эта команда кодирования отрицает команду кодирования UNTAGGED и по другому не влияет на кодирование.

24 Команда кодирования EMBED-VALUES

24.1 Общие положения

24.1.1 "EmbedValuesInstruction":

EmbedValuesInstruction ::=

EMBED-VALUES

TargetList

24.1.2 Продукционное правило "TargetList" определено в 13.2.

24.1.3 Эта команда кодирования позволяет первому компоненту типа последовательность (NOT UNTAGGED) обеспечить вставку строки символов перед первым элементом XML, после последнего элемента XML и между XML элементами, которые формируют результат кодирования "ExtendedXMLValue" типа последовательность.

24.1.4 Если окончательная команда кодирования USE-NIL также присутствует и OPTIONAL компонент, поддерживающий USE-NIL, отсутствует в конкретном абстрактном значении, тогда не будет никаких элементов XML для компонентов типа последовательность и не будет строки символов, обеспечивающих это абстрактное значение. Иначе, для всех абстрактных значений число строк символов обязано быть на единицу больше, чем число элементов в результате кодирования типа последовательность. Некоторые или все строки символов могут быть пустыми.

24.2 Ограничения

24.2.1 Тип ACH.1 не должен иметь этой окончательной команды кодирования, если только это не тип последовательность. Первый компонент последовательности SEQUENCE OF UTF8String не должен быть отмечен как OPTIONAL или DEFAULT.

24.2.2 Не должно быть окончательной команды кодирования UNTAGGED (см. раздел 31), примененной к типу последовательность-из или к компоненту последовательность-из.

24.2.3 Не должно быть окончательной команды кодирования UNTAGGED, применяемой к любому компоненту типа последовательность, который является типом character-encodable.

24.2.4 Не должно быть окончательной команды кодирования UNTAGGED примененной к какому-либо компоненту типа последовательность, который является посимвольно кодируемым типом.

24.2.5 Если тип последовательность имеет окончательную команду кодирования USE-NIL, то компонент OPTIONAL, поддерживающий команду кодирования USE-NIL, не должен быть посимвольно кодируемым типом (см. также 32.2.4).

24.2.6 Ни один из компонентов последовательности не должен быть отмечен DEFAULT, если только они не имеют окончательной команды кодирования ATTRIBUTE. Если есть компоненты типа SEQUENCE или SET (на любой глубине), то возможно создание элементов в "ExtendedXMLValue" с использованием UNTAGGED — непосредственных дочерних элементов типа последовательность. Они не должны быть отмечены как DEFAULT.

24.2.7 Тип последовательность должен быть ограничен таким способом что:

а) если тип имеет окончательную команду кодирования USE-NIL и OPTIONAL компонент, поддерживающий USE-NIL, отсутствует, число повторений компонентов последовательность-из обязано быть нулем;

б) иначе, число повторений компонентов последовательность-из в каждом абстрактном значении равно единице плюс число элементов XML в "ExtendedXMLValue" типа последовательность, определенного после применения всех окончательных команд кодирования к другим компонентам последовательности и игнорирования первого компонента.

Примечание — Рекомендуется выражать ограничение типа последовательность как:
(CONSTRAINED BY
{/* Должно соответствовать ИСО/МЭК 8825-4, раздел 25 */})

24.2.8 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

24.2.9 Тип с этой окончательной командой кодирования не должен иметь окончательной команды кодирования UNTAGGED.

Примечание — Нижеперечисленные окончательные команды кодирования не должны появляться вместе с этой окончательной командой кодирования потому, что их применение к типу запрещается: ANY-ATTRIBUTES, ANY-ELEMENT, ATTRIBUTE, BASE64, DECIMAL, LIST, TEXT, USENUMBER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE.

24.2.10 В "TargetList" не должно быть уточняющей информации.

24.3 Действие на результат кодирования

24.3.1 Кодер должен сначала продуцировать частичный результат "ExtendedXMLValue" кодирования типа включающая последовательность, игнорируя первый компонент. Это должно модифицировать результат кодирования, как определено в следующих подразделах.

Примечание — Значения UTF8String, которые вставляются, могут быть "пустыми".

24.3.1.1 Первое значение UTF8String должно быть вставлено в последовательность-из (см. 24.3.1.6) в начале результата частичного кодирования перед начальным тегом первого XML элемента (если есть хотя бы один).

24.3.1.2 Каждое последующее значение UTF8String (если есть хотя бы одно) должно быть вставлено между завершающим тегом XML элемента и начальным тегом следующего XML элемента, начиная от первого элемента к последнему.

Примечание — Вышеупомянутое подразумевает, что в каждый из этих элементов не вставлено UTF8String значение, даже если они имеют дочерние элементы.

24.3.1.3 Последнее значение UTF8String (если есть одно) должно быть вставлено в конец результата частичного кодирования после завершающего тега последнего XML элемента.

24.3.1.4 Если XML элементы отсутствуют в результате частичного кодирования, и есть также окончательная команда кодирования DEFAULT-FOR-EMPTY (см. 22), примененная к типу последовательность, и значение первой (и только) UTF8String в последовательности-из идентично "значению" "Value", специфицированному в команде кодирования DEFAULT-FOR-EMPTY, то кодер может опционально закодировать UTF8String как пустую строку (но см. 24.3.1.6).

24.3.1.5 Если XML элементы отсутствуют в результате частичного кодирования и есть окончательная команда кодирования DEFAULT-FOR-EMPTY, примененная к типу последовательность, и результат кодирования — пуст, декодер должен интерпретировать это как результат кодирования для "значения" "Value", специфицированного в команде кодирования DEFAULT-FOR-EMPTY и назначить это абстрактное значение первой (и только) UTF8String в последовательности-из (но см. 24.3.1.6).

Примечание — Это означает, что значение без XML элементов и с единственным пустым значением UTF8String не может быть закодировано. Тип последовательность обязан быть ограниченным, чтобы запретить такие значения (см. 22.2.5).

24.3.1.6 Если тип также имеет окончательную команду кодирования USE-NIL и OPTIONAL компонент отсутствует, то команда кодирования EMBED-VALUES не оказывает действие на результат кодирования. Если тип также имеет окончательную команду кодирования USE-NIL и OPTIONAL компонент присутствует, тогда применяется условие 24.3.1.4. Если декодер определяет, что OPTIONAL компонент присутствует по отсутствию атрибута идентификации пустого множества (или его присутствию со значением false), тогда применяется условие 24.3.1.5.

25 Команда кодирования GLOBAL-DEFAULTS

25.1 Общие положения

25.1.1 "GlobalDefaultsInstruction" :

```
GlobalDefaultsInstruction::=
        GLOBAL-DEFAULTS TargetList DefaultSetting
DefaultSetting::=
        ControlNamespace
        | MODIFIED-ENCODINGS
ControlNamespace::=
        CONTROL-NAMESPACE
        QuotedURI
        Prefix ?
```

25.1.2 Продукционное правило "TargetList" определено в 13.2 и должно быть "пустым" "empty".

25.1.3 "QuotedURI" и "Prefix" определены в 28.1.1.

25.1.4 Продукционное правило "ControlNamespace" определяет название управляющего пространства имен ("URI" в "QuotedURI") и рекомендуемый префикс для этого пространства имен. Если эта команда кодирования GLOBAL-DEFAULTS не представлена, управляющее пространство имен должно быть тем, какое определено в 15.9.

25.1.5 Использование MODIFIED-ENCODINGS продуцирует "ExtendedXMLValues", которые модифицированы в соответствии с 9.2.7 и 9.2.8.

25.2 Ограничения

25.2.1 Команда кодирования GLOBAL-DEFAULTS должна быть назначена только в разделе управления кодированием, и ей не должны предшествовать никакие команды кодирования, кроме других команд кодирования GLOBAL-DEFAULTS,.

25.2.2 Каждая из альтернатив GLOBAL-DEFAULTS должна использоваться только однажды в любом разделе управления кодированием.

25.2.3 GLOBAL-DEFAULTS MODIFIED-ENCODINGS, если присутствует, должна быть первой командой в разделе управления XER кодированием в модуле ACH.1.

25.3 Действие на результат кодирования

25.3.1 Применение MODIFIED-ENCODINGS требует, чтобы кодирование было модифицировано, как определено в 9.2.7 и 9.2.8.

25.3.2 Управляющее пространство имен, используемое для полного документа XML, должно быть управляющим пространством имен, назначенным типу ACH.1, кодирование которого формирует корневой элемент этого XML документа.

26 Команда кодирования LIST

26.1 Общие положения

26.1.1 "ListInstruction" :

```
ListInstruction::=
        LIST
        TargetList
```

26.1.2 Продукционное правило "TargetList" определено в 13.2.

26.1.3 Эта команда кодирования требует, чтобы "ExtendedXMLSequenceOfValue" или "ExtendedXMLSetOfValue" типа последовательность-из или набор-из (см. 16.7) были "ExtendedXMLListValue", продуцирующие разделенный пробелами перечень значений компонента последовательность-из или набор-из.

Примечание — Обычно эту команду кодирования назначают SEQUENCE OF INTEGER, которой команда кодирование ATTRIBUTE (см. раздел 19) также назначается.

26.2 Ограничения

26.2.1 Тип, которому назначается эта команда кодирования, должен быть последовательность-из или набор-из.

26.2.2 Компонент последовательность-из или набор-из:

а) должен быть посимвольно кодируемого типа;

б) должен быть таким, чтобы для всех его абстрактных значений имелся, по крайней мере, один результат кодирования "ExtendedXMLValue" (принимая во внимание все опции кодера), не "пустой" "empty" не содержащий "пробел с переходом" (см. 7.1.5).

Примечание 1 — Вышеупомянутые ограничения подразумевают, что компонент не может самостоятельно быть типа последовательность-из или набор-из с командой кодирования LIST, или содержать вложенный тип последовательность-из или набор-из с командой кодированием LIST на любой глубине.

Примечание 2 — Вышеупомянутые ограничения будут удовлетворены, если тип компонента последовательность-из или набор-из является целочисленным, действительным, типом идентификатора объекта, типом относительного идентификатора объекта или такими полезными типами как GeneralizedTime и UTCTime. Они будут также удовлетворены, если это тип строка символов ограниченный так, чтобы всегда имелся, по крайней мере, один символ в строке символов и ни одно из его значений не содержало символ "пробел" "white-space".

Примечание 3 — Признано, что некоторые инструменты АСН.1 не в состоянии статически проверить, что вышеупомянутые правила удовлетворены, но кодер, соответствующий правилам, не должен производить кодирование, которое нарушает ограничение (б).

26.2.3 Тип с этой окончательной командой кодирования не должен иметь окончательной команды кодирования ANY-ATTRIBUTES.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не могут появляться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ELEMENT, BASE64, DECIMAL, EMBED-VALUES, TEXT, USE-NIL, USE-NUMBER, USEORDER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE.

26.2.4 В "TargetList" не должно быть уточняющей информации.

26.3 Действие на результат кодирования

26.3.1 Эта команда кодирования действует только на результат кодирования типа, к которому применена.

26.3.2 Продукционные правила "ExtendedXMLSequenceOfValue" или "ExtendedXMLSetOfValue" (см.16.7) должны быть альтернативой "ExtendedXMLListValue". "ExtendedXMLListValue" есть:

```
ExtendedXMLListValue::=
        empty
        | CharacterEncodableValue ExtendedXMLListValue
```

26.3.3 Должен быть "пробел с переходом" (см. 7.1.5) между каждой парой смежных "CharacterEncodableValue" в "ExtendedXMLListValue".

26.3.4 "CharacterEncodableValue" определен в 19.3.3. Каждый "CharacterEncodableValue" должен закодировать значение из компонента последовательность-из или набора-из.

26.3.5 Порядок, в котором значения "CharacterEncodableValue"s появляются в "ExtendedXMLListValue", должен быть тот же самый, в котором соответствующие значения "ExtendedXMLValue"s появились бы в "ExtendedXMLSequenceOfValue" или "ExtendedXMLSetOfValue", если окончательная команда кодирования LIST отсутствовала.

26.3.6 Значения "CharacterEncodableValue"s в "ExtendedXMLListValue" не должны быть "пустыми" "empty" и не должны содержать "пробел с переходом" (см. 7.1.5).

Примечание — Подраздел 26.2.2 (б) гарантирует, что это возможно, но 26.3.4 может ограничить опции кодера устройства.

27 Команда кодирования NAME

27.1 Общие положения

27.1.1 "NameInstruction":

```
NameInstruction:: =
        NAME
        TargetList
        AS
        NewNameOrKeyword
```

```
NewNameOrKeyword::=
        NewName
        | Keyword
NewName::=
        RestrictedCharacterStringValue
Keyword::=
        CAPITALIZED
        | UNCAPITALIZED
        | UPPERCASED
        | LOWERCASED
```

27.1.2 Продукционное правило "TargetList" определено в 13.2.

27.1.3 Эта команда кодирования имеет пять самостоятельных назначений:

а) изменять связанное имя тега, имя атрибута или значение атрибута возможной идентификации типа («новое имя» "NewName" без «уточняющей информации» "QualifyingInformation" в «перечне целей» "TargetList") цели;

б) изменять регистр (или регистр первой буквы) имени связанного тега, имени атрибута, или значения атрибута возможной идентификации типа («ключевое слово» "Keyword" без «уточняющей информации» "QualifyingInformation" в «перечне целей» "TargetList") цели (ей);

в) изменять имя элемента, применяемое в теге «пустой элемент» обычно (как определено в ИСО/МЭК 8824-1) получаемого из специфицированного идентификатора, используемого в определении типа («новое имя» "NewName" с «уточняющей информацией» "QualifyingInformation" в «перечне целей» "TargetList", которая не является ALL) цели;

г) изменять регистр (или регистр первой буквы) имени элемента, применяемого в теге «пустого элемента», обычно получаемого из специфицированного идентификатора, используемого в определении типа («ключевое слово» "Keyword" с «уточняющей информацией» "QualifyingInformation" в «перечне целей» "TargetList", которая не является ALL) цели (ей);

д) изменять регистр (или регистр первой буквы) имен элементов, применяемых при кодировании "ExtendedXMLValue", получаемом из любого идентификатора, используемого в определении типа («ключевое слово» "Keyword" с «уточняющей информацией» "QualifyingInformation" в «перечне целей» "TargetList", которая является ALL) цели (ей).

Примечание 1 — «Новое имя» "NewName" может применяться для изменения имен, используемых при EXTENDED-XER кодировании, получаемых из идентификаторов или ссылок на типы, но является исключительно полезным, если новое имя может быть использовано, в первую очередь, как идентификатор ACH.1 или ссылка на тип. Таким образом, назначение команды кодирования NAME — это продуцирование требуемого XML элемента или имен атрибутов, когда они не разрешены из-за правил ACH.1 относительно регистра первой буквы идентификаторов или названий ссылки на тип, а также когда правила ACH.1 для различных идентификаторов в конструкциях последовательность, набор и выбор предотвращают желательное кодирование XML.

Примечание 2 — Использование ALL, IN ALL, AS, CAPITALIZED для изображения прописными буквами всех идентификаторов в модуле, может быть особенно полезным для обеспечения общего стиля использования начальных заглавных букв.

Примечание 3 — Если команда кодирования NAME назначена с использованием цели, идентифицированной «идентификатором» "identifier" или «ссылкой на тип» "typereference", это затрагивает имя, используемое в EXTENDED-XER кодировании, но не затрагивает имени, которое используется, чтобы идентифицировать ту же самую цель в последующих командах XER кодирования.

27.1.4 Значение "RestrictedCharacterStringValue" определено в ИСО/МЭК 8824-1, раздел 37.

27.2 Ограничения

27.2.1 "NewName" не должно использоваться, если "QualifyingInformation" будет ALL.

27.2.2 Команда кодирования NAME с "QualifyingInformation" должна назначаться только для следующих определений типа:

а) определение типа булевский;

б) определение типа строка битов с поименованными битами;

в) определение типа перечислимый;

г) определение типа целочисленный с поименованными значениями.

27.2.3 Значение "RestrictedCharacterStringValue" в "NewName" при использовании в команде кодирования NAME должно быть или "NCName", определенное в W3C XML Namespaces (раздел 2, продукционное

правило 4), или пустая строка символов. Она не должна быть пустой строкой символов, если только команда кодирования NAME не применяется к альтернативе типа выбора с окончательной командой кодирования USE-UNION.

Примечание 1 — Это требование из "W3C XML Namespaces": "NCName" не начинается с символов, которые будучи заглавными, представляют собой "XML".

Примечание 2 — Продукционное правило "NewNameOrKeyword" (и, следовательно, продукционное правило "NewName") также используется в разделе 30. Вышеупомянутые ограничения на "RestrictedCharacterStringValue" не применяются к использованию в разделе 30.

27.2.4 Если есть команда кодирования GLOBAL-DEFAULTS с ключевым словом MODIFIED-ENCODINGS, то не должно быть никакой "QualifyingInformation" в "TargetList".

Примечание — это объясняется тем, что теги пустого элемента не используются в данном случае. Команда кодирования TEXT может использоваться вместо этого для изменения результатов кодирования индивидуальных значений типа.

27.2.5 Эта команда кодирования не должна использоваться в качестве команды кодирования, снабженной префиксом в комбинации ни с одной из следующих команд кодирования с префиксом: ANY-ATTRIBUTES, ANY-ELEMENT или UNTAGGED, чтобы избежать дезориентации считывателя.

27.3 Действие на результат кодирования

27.3.1 Если тип, к которому применена эта команда кодирования, имеет окончательную команду кодирования ATTRIBUTE, "AttributeName" (который является в этом случае "IdentifierOrModifiedIdentifier") "атрибута" "Attribute" (см. 19.3.3) должно быть альтернативой "QualifiedOrUnqualifiedName", как определено в 27.3.3 — 27.3.6.

27.3.2 Если тип, к которому применена эта команда кодирования, не имеет окончательной команды кодирования ATTRIBUTE, тогда имя тега включающего элемента (который является "TagName" — см. 16.5.1) должно быть альтернативой "QualifiedOrUnqualifiedName", как определено в 27.3.3 — 27.3.6.

27.3.3 Альтернативы "IdentifierOrModifiedIdentifier" и "QualifiedOrUnqualifiedName" должны использоваться. "UnprefixedName" в "QualifiedOrUnqualifiedName" должен быть "идентификатором" "identifier" компонента, модифицированного в соответствии с "NewNameOrKeyword" как определено ниже.

27.3.4 Если альтернатива "NewName" будет использоваться, то "UnprefixedName" должен быть заменен "NewName".

27.3.5 Если альтернатива "Keyword" будет использоваться, то "UnprefixedName" должен быть модифицирован, как определено в подразделах 27.3.5.1 — 27.3.5.4.

27.3.5.1 Если "Keyword" — CAPITALIZED, то, если первый символ "UnprefixedName"— строчная буква, этот символ должен быть заменен прописным эквивалентом, иначе название не затронуто.

27.3.5.2 Если "Keyword" — UNCAPITALIZED, то, если первый символ "UnprefixedName" — прописная буква, этот символ должен быть заменен строчным эквивалентом, иначе название не затронуто.

27.3.5.3 Если "Keyword" — UPPERCASED, то все символы "UnprefixedName", которые являются строчными буквами, будут заменены их прописными эквивалентами. Другие символы неизменны.

27.3.5.4 Если "Keyword" — LOWERCASED, то все символы "UnprefixedName", которые являются прописными буквами, будут заменены их строчными эквивалентами. Другие символы неизменны.

27.3.6 "QualifiedOrUnqualifiedName" будет именем пространства уточненных имен только в случае, если "Type" имеет окончательную команду кодирования NAMESPACE.

28 Команда кодирования NAMESPACE

28.1 Общие положения

28.1.1 "NamespaceInstruction":

```
NamespaceInstruction::=
        NAMESPACE
        TargetList
        NamespaceSpecification ?
NamespaceSpecification::=
        AS
        QuotedURI
        Prefix ?
```

Prefix::=
PREFIX
QuotedNCName
QuotedURI::=
""" & URI & """
QuotedNCName::=
""" & NCName & """

28.1.2 Продукционное правило "TargetList" определено в 13.2.

Примечание — Самое общее использование этой инструкции кодирования — NAMESPACE ALL.

28.1.3 Эта команда кодирования позволяет имени пространства имен (namespace) и рекомендованному префиксу пространства имен (namespace) быть назначенными цели (целям).

28.1.4 Продукционное правило "URI" не определено в настоящем Международном стандарте, но оно состоит из символов, которые идентифицируют Унифицированный Идентификатор Ресурса (Uniform Resource Identifier — URI). Синтаксис (и семантика) URI определены в IETF RFC 2396, и начинаются с имени схемы URI. Для распределения имен пространства имен с командой кодирования NAMESPACE может использоваться любая схема URI.

28.1.5 Если "NamespaceSpecification" отсутствует, то значение по умолчанию с рекомендованным установленным "Prefix" назначается ссылочному модулю "modulereference" и "URI" устанавливается в следующем виде:

а) схема URI (см. IETF RFC 2396) должна быть urn;

б) идентификатор пространства имен URN (см. IETF RFC 2141) должен быть oid;

в) Специфическая Строка Пространства Имен URN (см. IETF RFC 2141) должна быть "Отличительным Идентификатором" "DefinitiveIdentifier" модуля, выраженного как "Значение Идентификатора XML Объекта" "XMLObjectIdentifierValue" (см. IETF RFC 3061).

28.1.6 Пример: При значении идентификатора объекта {iso standard 1564 modules(0) basic(1)} URI был бы строкой символов "urn:oid:1.0.1564.0.1".

28.1.7 Продукционное правило "NCName" определено в Пространстве имен W3C XML (W3C XML Namespaces), разделе 2, продукционное правило 4, и не будет начинаться с символов, которые являются заглавными "XML".

Примечание — Это требование, предписанное Рекомендацией W3C XML Пространство имен.

28.2 Ограничения

28.2.1 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

28.3 Действие на результат кодирования

28.3.1 Имя уточненного пространства имен (Namespace-qualified) может потребоваться для имени связанного тега, для имени атрибута, или для значения атрибута идентификации типа. Имя уточненного пространства имен требуется, если тип, генерирующий название, является окончательной командой кодирования NAMESPACE.

28.3.2 "QualifiedOrUnqualifiedName":
QualifiedOrUnqualifiedName::=
QualifiedName |
UnqualifiedName
QualifiedName::=
PrefixedName |
UnprefixedName
UnqualifiedName::=
UnprefixedName
PrefixedName::=
DeclaredPrefix & ":" & UnprefixedName
UnprefixedName::= NCName
DeclaredPrefix::= NCName

28.3.3 Кодирование имени уточненного пространства имен требует:

а) использования альтернативы "ИмяСПрефиксом" "PrefixedName" для "УточненногоИмени" "QualifiedName" с дополнением к элементам XML добавочных признаков, обеспечивающих декларации пространства имен (как определено в W3C XML Namespaces);

б) использования альтернативы "ИмяБезПрефикса" "UnprefixedName" для "Уточненного Имени" "QualifiedName" с дополнением к элементам XML добавочных признаков, обеспечивающих декларации пространства имен по умолчанию (как определено в W3C XML Namespaces).

28.3.4 Выбор этих двух механизмов и элементов XML, к которым добавляются признаки декларации пространства имен, являются опцией кодера.

Примечание 1 — Рекомендация W3C XML Namespaces определяет, что объявленное по умолчанию пространство имен имеет в своей области действия только имя элемента, в котором оно объявлено (и имена дочерних элементов), но не атрибуты элементов или дочерних элементов.

Примечание 2 — Рекомендуется, но не требуется, использование рекомендованного префикса в команде кодирования NAMESPACE.

Примечание 3 — Использование рекомендованного префикса может быть неприемлемым, если команды кодирования NAMESPACE с другими именами пространства имен namespace, но с теми же самыми рекомендованными префиксами присутствуют в модуле.

29 Команда кодирования PI-OR-COMMENT

29.1 Общие положения

29.1.1 "PIOrCommentInstruction":

```
PiOrCommentInstruction::=
        PI-OR-COMMENT
        TargetList
        AS
        RestrictedCharacterStringValue
        Position
Position ::=
        BEFORE-TAG
        | BEFORE-VALUE
        | AFTER-VALUE
        | AFTER-TAG
```

29.1.2 Продукционное правило "TargetList" определено в 13.2.

29.1.3 Эта команда кодирования вызывает определенные команды обработки XML и/или комментарии, которые будут вставлены перед или после "ExtendedXMLValue" или перед или после связанных тегов.

Примечание — Подраздел 9.2.5 разрешает кодеру (как выбор кодера) вставлять дополнительные XML команды обработки и XML комментарии.

29.1.4 "RestrictedCharacterStringValue" определен в ИСО/МЭК 8824-1, 37.

29.2 Ограничения

29.2.1 Значение "RestrictedCharacterStringValue" должно быть связано с одной или более строками символов, каждая из которых соответствует синтаксису команд обработки XML (Processing Instruction), определенному в W3C XML 1.0, 2.6, или синтаксису комментариев XML (Comment), определенному в W3C XML 1.0, 2.5, и определяет команды обработки и/или комментарии, которые должны быть вставлены в XML документ.

29.2.2 Тип ACH.1 не должен иметь окончательных команд кодирования UNTAGGED и PI-OR-COMMENT.

29.2.3 Тип этой окончательной команды кодирования не должен иметь окончательных команд кодирования ANY-ATTRIBUTES, ANY-ELEMENT, ATTRIBUTE или UNTAGGED.

29.2.4 В "TargetList" не должно быть уточняющей информации.

29.3 Действие на результат кодирования

29.3.1 Если позиция ("Position") — BEFORE-TAG, то команды обработки и/или комментарии должны быть вставлены перед связанным начальным тегом или тегом пустого элемента. Если начальный тег или тег пустого элемента — начало некоторого включающего "ExtendedXMLValue", тогда любые команды обработки и/или комментарии, вставленные перед этим "ExtendedXMLvalue" (применение BEFORE-VALUE к соответствующему типу), будут предшествовать этим командам обработки и/или комментариям в XML документе.

29.3.2 Если позиция ("Position") — BEFORE-VALUE, то команды обработки и/или комментарии должны быть вставлены в начало "ExtendedXMLValue". Если "ExtendedXMLValue" начинается с тега, который

является связанным начальным тегом некоторых вложенных "ExtendedXMLValue", тогда любые команды обработки и/или комментарии, вставленные прежде, чем этот связанный начальный тег (применение BEFORE-TAG к соответствующему типу), должны следовать за этими командами обработки и/или комментариями в XML документе.

Примечание — В этом случае содержание связанных тегов никогда не пусто, и атрибут пустого элемента не может использоваться.

29.3.3 Если позиция ("Position") — AFTER-VALUE, то команды обработки и/или комментарии должны быть вставлены в конец "ExtendedXMLValue". Если "ExtendedXMLValue" заканчивается тегом, который является связанным конечным тегом некоторых вложенных "ExtendedXMLValue", тогда любые команды обработки и/или комментарии, вставленные после связанного конечного тега (применение AFTER-TAG к соответствующему типу), будут предшествовать этим командам обработки и/или комментариям в XML документе.

Примечание — В этом случае содержание связанных тегов никогда не пусто, и тег пустого элемента не может использоваться.

29.3.4 Если позиция ("Position") — AFTER-TAG, то команды обработки и/или комментарии должны быть вставлены после связанного конечного тега или тега пустого элемента. Если конечный тег или тег пустого элемента завершает некоторое приложение "ExtendedXMLValue", тогда любые команды обработки и/или комментарии должны быть вставлены после этого "ExtendedXMLvalue" (применение AFTER-VALUE к соответствующему типу) и должны следовать за этими командами обработки и/или комментарии в XML документе.

30 Команда кодирования TEXT

30.1 Общие положения

30.1.1 "TextInstruction":

TextInstruction::=
    TEXT
    TargetList
    TextToBeUsed ?
TextToBeUsed::=
    AS
    NewNameOrKeyword

30.1.2 Продукционное правило "TargetList" определено в 13.2.

30.1.3 Цель этой команды кодирования:

а) в отсутствие GLOBAL-DEFAULTS MODIFIED-ENCODINGS позволить значениям булевского типа, перечислимого типа, типа строка битов с поименованными битами и целочисленных с поименованными числами, быть закодированными как строки символов, а не как теги пустого элемента;

б) в присутствии GLOBAL-DEFAULTS MODIFIED-ENCODINGS позволить изменять строки символов, которые используются для значений булевских типов, перечислимых типов, строк битов с поименованными битами и целочисленных с поименованными значениями.

30.1.4 "NewNameOrKeyword" определен в разделе 27. "NewName" в "NewNameOrKeyword" будет содержать, по крайней мере, один символ.

30.2 Ограничения

30.2.1 Эта команда кодирования должна быть назначена только на следующие типы с уточняющей информацией, определяющей один или более идентификаторов, используемых в определении типа (true или false для булевского типа):

а) определение булевского типа;

б) определение типа строка битов с поименованными битами;

в) определение перечислимого типа;

г) определение целочисленного типа с названными значениями.

30.2.2 Окончательные строки символов, используемые для значений типа, которому назначена эта команда кодирования, должны различаться.

30.2.3 "NewName" в "NewNameOrKeyword" не должен использоваться, если "QualifyingInformation" имеет значение ALL. Ограничение, приведенное в 27.2.3, не применимо к этому использованию "NewNameOrKeyword".

30.2.4 В отсутствие команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS набор окончательных команд кодирования TEXT для типа не должен производить кодирования текста для некоторых абстрактных значений и кодирования пустых элементов для других абстрактных значений.

Примечание — Если есть команда кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS, то все кодирования — кодирования текста.

30.2.5 Если команда кодирования TEXT применена к типу строка битов с поименованными битами и "NewName" используется, "NewName" не должно содержать "пробел с переходом" (см. 7.1.5) и не будет начинаться с "0" (DIGIT ZERO) или "1" (DIGIT ONE).

30.2.6 Тип с этой окончательной командой кодирования не должен иметь окончательной команды кодирования USE-NUMBER.

Примечание — Нижеперечисленные окончательные команды кодирования не должны сочетаться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, ANY-ELEMENT, BASE64, DECIMAL, EMBED-VALUES, LIST, USE-NIL, USEORDER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE.

30.2.7 "QualifyingInformation" должна всегда присутствовать.

30.3 Действие на результат кодирования

30.3.1 Один из следующих пяти подразделов (30.3.2 — 30.3.6) применяется.

30.3.2 Если тип — не строка битов с поименованными битами и "TextToBeUsed" отсутствует, результатом кодирования "ExtendedXMLValue" каждого значения, на которую ссылается уточняющая информация для этой команды, должна быть строка символов, содержащая символы идентификатора (или будет true или false в случае булевских типов). Для целочисленных типов с поименованными значениями должны быть использованы идентификаторы или соответствующие значения (как выбор кодера).

30.3.3 Если тип — строка битов с поименованными битами и "TextToBeUsed" отсутствует, строка символов, идентичная идентификатору бита должна представлять бит, когда это установлено. Каждое абстрактное значение должно быть закодировано как сцепление (возможно пустое) этих строк символов для всех битов, которые установлены и разделены "пробелом с переходом" (см. 7.1.5).

30.3.4 Если тип — не строка битов с поименованными битами и "TextToBeUsed" присутствует, то следующие подразделы применяются (но см. 30.3.5).

30.3.4.1 Если применяется альтернатива "NewName", строка символов использованная для кодирования значения, идентифицированного "QualifyingInformation", — "NewName". Каждое появление символов "<", ">" и "&" в "NewName" будет заменено одной из управляющих последовательностей "<", ">", и "&" соответственно, или управляющей последовательностью формы "&#n;" или "&#xn;", определенной в ИСО/МЭК 8824-1, 11.15.8.

30.3.4.2 Если применяется альтернатива "Keyword", строка символов, использованная для кодирования значений типа, — имя идентификатора, модифицированное, как определено ниже.

30.3.4.3 Если "Keyword" — CAPITALIZED, то первый символ имени заменяется заглавным эквивалентом, иначе имя не меняется.

30.3.4.4 Если "Keyword" — UNCAPITALIZED, то имя неизменно.

30.3.4.5 Если "Keyword" — UPPERCASED, то все символы имени, которые являются строчными буквами, заменяются их прописными эквивалентами. Другие символы неизменны.

30.3.4.6 Если "Keyword" — LOWERCASED, то все символы имени, которые являются прописными буквами, заменяются их строчными эквивалентами. Другие символы неизменны.

30.3.5 Если тип — целочисленный тип с поименованными значениями, строки символов, созданные с применением подразделов 30.3.4.1 — 30.3.4.6, будут использоваться вместо идентификаторов. Строки символов или соответствующие значения должны использоваться (как выбор кодера).

30.3.6 Если тип — строка битов с поименованными битами и "TextToBeUsed" присутствует, подразделы 30.3.4.1 — 30.3.4.6 будут применены к каждому идентификатору бита, чтобы создать строку символов, которая представляет бит, когда он установлен. Значение строки битов должно быть закодировано как сцепление (возможно пустое) этих строк символов для всех битов, которые установлены и разделены "пробелом с переходом" (см. 7.1.5).

31 Команда кодирования UNTAGGED

31.1 Общие положения

31.1.1 "UntaggedInstruction" :

UntaggedInstruction::=

UNTAGGED

TargetList

31.1.2 Продукционное правило "TargetList" определено в 13.2.

31.1.3 (Инструкция) неофициальное описание эффекта UNTAGGED на конструкторы АСН.1 приведено в приложении Б.

В этом разделе дано вводное руководство, иллюстрирующее некоторые эффекты использования UNTAGGED.

31.1.4 Использованная (возможно, неоднократно и во вложеном виде) в соединении с последовательностью, набором, выбором, последовательностью-из и набором-из позволяет определять почти произвольный образец элементов XML. В результате удаляется начальный тег XML, предшествующий значению "ExtendedXMLValue" в "Type", к которому это значение применено, и конечный тег XML, который следует за этим значением. Остаются элементы XML, обычно содержавшиеся между этими тегами, становящимися частичным XML содержанием.

31.1.5 Данная команда, примененная к типу выбора как компоненту последовательности или набора, определяет включение в месте последовательности (набора) точно одной из альтернатив типа выбора (или ни одной, если типом выбора является OPTIONAL компонент). Идентификатор типа выбора не появляется в кодировании. Некоторые альтернативы для типа выбора могут быть XML элементами, но другие могут быть частичным XML контентом, содержащим почти произвольный шаблон множественных элементов в результате использованиия UNTAGGED при определении тех альтернатив.

31.1.6 Примененная к типу последовательность-из как компоненту последовательности или набора, она определяет включение в месте последовательности (набора) указанного или произвольного числа повторений компонента последовательность-из (которые могут создать единичный XML элемент, или создать частичный XML контент, если он сам является UNTAGGED).

31.1.7 Примененная к типу последовательности (или набора) или к типу последовательность-из (или набор-из) как альтернатива типа выбора, она позволяет альтернативе состоять из частичного XML контента, который является "ExtendedXMLValue" последовательности, набора, последовательности-из или набора-из.

31.1.8 Отдельная функция UNTAGGED, будучи примененной к типу посимвольно-кодируемый, должна позволить символьному контенту появиться в кодировании последовательности без тегов вокруг этого контента. Это использование ограничено компонентами последовательности, которые сами не являются нетеговыми.

Примечание — Это ограничение необходимо, чтобы упростить правила, что обеспечит легкое и однозначное декодирование.

31.2 Ограничения

31.2.1 Во всех случаях использования включающий тип должен быть типом последовательности, набора, выбора, последовательности-из или набора-из.

31.2.2 Если тип будет посимвольно-кодируемым, то включающий тип должен быть типом последовательности без окончательной команды кодирования UNTAGGED. Тип не должен быть отмечен OPTIONAL или DEFAULT. Все другие компоненты включающего типа последовательности (если таковые существуют) должны иметь окончательные команды кодирования ATTRIBUTE или ANY-ATTRIBUTES.

31.2.3 Если тип не будет типом посимвольно-кодируемым, то это должен быть тип последовательности, набора, выбора, последовательности-из, набора-из, строки октетов или строки битов с содержащимся "Type" без ENCODED BY или открытого типа.

Примечание — Приложение Б содержит руководство, которое может гарантировать, что использование этой инструкции кодирования не приводит к двусмысленности.

31.2.4 Эта команда кодирования не должна быть применена к типу, который имеет пустой результат кодирования "ExtendedXMLValue" для одного из его абстрактных значений, если тип используется как:

а) компонент типа последовательности или набора с OPTIONAL или DEFAULT;

б) компонент типа последовательности-из или набора-из;

в) альтернатива для типа выбора, если другая альтернатива для того же самого типа выбора имеет пустой результат кодирования "ExtendedXMLValue" для одного из его абстрактных значений и имеет окончательную команду кодирования UNTAGGED.

Пример: тип, который является типом последовательности со всеми его OPTIONAL компонентами, имеет абстрактное значение с пустым результатом кодирования "ExtendedXMLValue", как и тип последовательность-из, где нулевые повторения разрешены.

31.2.5 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

31.2.6 Тип с этой окончательной командой кодирования не должен иметь ни одной из перечисленных окончательных команд кодирования ANY-ATTRIBUTE, ANY-ELEMENT, ATTRIBUTE, DEFAULT-FOR-EMPTY, EMBED-VALUES, PI-OR-COMMENT, USE-NIL, USEORDER или USE-TYPE.

31.2.7 В "TargetList" не должно быть никакой уточняющей информации.

31.3 Действие на результат кодирования

31.3.1 Если тип закодирован как тип высшего уровня, то эта команда кодирования должна игнорироваться.

31.3.2 Если включающий тип — тип выбора, "ExtendedXMLChoiceValue" (см. 16.5.1) для этой альтернативы для вложенного типа должен быть "ExtendedXMLValue" альтернативы (вторая альтернатива в продукционных правилах "ExtendedXMLChoiceValue").

Примечание — "ExtendedXMLValue" для альтернативы может быть единственным XML элементом или может быть частичным XML контентом, состоящим из множественных XML элементов.

31.3.3 Если включающий тип — тип последовательности или набора, "ExtendedXMLNamedValue" (см. 16.6) для этого компонента вложенного типа будет заменен на "ExtendedXMLValue" компонента (вторая альтернатива в продукционных правилах "ExtendedXMLNamedValue").

Примечание — Этот "ExtendedXMLValue" может быть единственным XML элементом или может быть частичным XML контентом, состоящим из множественных XML элементов.

31.3.4 Если вложенный тип — тип последовательности-из или набора-из, "ExtendedXMLDelimitedItem" (если используется — см. 16.7) каждого повторения должно быть заменено "ExtendedXMLValue", вложенным в "ExtendedXMLDelimitedItem".

Примечание 1 — Невозможно использовать UNTAGGED, если только GLOBAL-DEFAULTS MODIFIED-ENCODINGS не была включена в раздел управления кодированиием, когда "ExtendedXMLValueList" не разрешается (см. 17.7.2).

Примечание 2 — Этот "ExtendedXMLValue" может быть единственным XML элементом или может быть частичным XML контентом, состоящим из множественных XML элементов.

31.3.5 Если тип — строка октетов или строка битов с содержащимся "Type" без ENCODED BY, или открытым типом, "ExtendedXMLValue" должен быть "ExtendedXMLTypedValue" (не "xmlhstring" или "XMLBase64String").

Примечание — Такие типы не соответствуют определению посимвольно-кодируемого типа (см. 3.2.2.2). Подраздел 31.3.5 подразумевает, что когда они имеют окончательную команду кодирования UNTAGGED, они всегда кодируются как XML элементы.

32 Команда кодирования USE-NIL

32.1 Общие положения

32.1.1 "UseNilInstruction":

UseNilInstruction::=

USE-NIL

TargetList

32.1.2 Продукционное правило "TargetList" определено в 13.2.

32.1.3 Эта команда кодирования обеспечивает оптимизированное EXTENDED-XER кодирование для последовательности с единственным OPTIONAL компонентом, все другие компоненты которого (если они существуют) имеют окончательную команду кодирования ATTRIBUTE или ANY-ATTRIBUTE, которой, возможно, предшествует начальный тип последовательности-из, поддерживающий USE-ORDER (см. 34).

32.1.4 В отсутствие этой команды кодирования опциональный компонент может быть закодирован следующим образом:

а) (вариант «не пропущенный, но пустой»), если компонент присутствует в абстрактном значении с абстрактным значением, которое имеет пустой результат кодирования "ExtendedXMLValue", тогда "ExtendedXMLNamedValue" для компонента присутствует в XML документе обычно как тег пустого элемента (или со смежными начальным и конечным тегами);

б) (вариант «пропущен»), если компонент отсутствует в абстрактном значении, "ExtendedXMLNamedValue" не представлен;

в) (вариант «не пропущенный и не пустой»), если компонент присутствует в абстрактном значении с абстрактным значением, которое не имеет пустого результата кодирования, тогда "ExtendedXMLNamedValue" для компонента представлен с непустым содержанием.

32.1.5 Использование USE-NIL требует, чтобы отсутствие дополнительного компонента (32.1.4 б) было сообщено включением нулевого идентификационного атрибута идентификации с именем "nil" и значением "true" или "1".

32.1.6 В случаях 32.1.4 а и б, нулевой признак идентификации может быть опущен (как выбор кодера) или он может присутствовать со значением "false" или "0". Опциональный компонент должен быть закодирован исключением связанных тегов.

32.2 Ограничения

32.2.1 Команда кодирования USE-NIL должна быть только назначена для типа последовательность, который имеет OPTIONAL компонент без окончательной команды кодирования ATTRIBUTE. Все другие компоненты типа последовательность, если таковые имеются, должны иметь окончательную команду кодирования ATTRIBUTE или ANY-ATTRIBUTE или должны быть компонентами последовательность-из, поддерживающими команды кодирования USE-ORDER или EMBED-VALUE, которые являются также окончательными командами кодирования для типа последовательность.

32.2.2 Тип последовательность не должен иметь окончательной команды кодирования UNTAGGED.

32.2.3 Компонент OPTIONAL не должен иметь ни одной окончательной команды кодирования ANY-ELEMENT, ANY-ATTRIBUTE, DEFAULT-FOR-EMPTY, EMBED-VALUE, PI-OR-COMMENT, UNTAGGED, USE-NIL, USE-ORDER или USETYPE.

Примечание — Кроме UNTAGGED, упомянутые выше команды кодирования не могут быть применены к типу, который имеет окончательную команду кодирования UNTAGGED.

32.2.4 Если компонент OPTIONAL не посимвольно кодируемого типа, то это должен быть тип последовательность, набор, выбор, последовательность-из, набор-из, открытый тип или тип строка октетов, или строка битов, содержащая тип "Type" и без ENCODED BY.

32.2.5 Если компонент OPTIONAL будет типом последовательность, то ни один из его компонентов не должен иметь окончательной команды кодирования ATTRIBUTE или ANY-ATTRIBUTE.

32.2.6 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

32.2.7 Тип с этой окончательной командой кодирования не должен иметь ни одной из окончательных команд кодирования UNTAGGED или USE-QNAME.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не должны появляться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, ANY-ELEMENT, ATTRIBUTE, BASE64, DECIMAL, LIST, TEXT, USENUMBER, USE-TYPE, USE-UNION, WHITESPACE.

32.2.8 В "TargetList" не должно быть уточняющей информации.

32.3 Действие на результат кодирования

32.3.1 Если компонент OPTIONAL отсутствует (32.1.4 б), то нулевой атрибут идентификации пустого множества с названием "nil" и значением "true" или "1" должен быть добавлен к "AttributeList" включающего элемента.

32.3.2 Если компонент OPTIONAL присутствует (32.1.4 а и в), атрибут идентификации пустого может быть опущен (как выбор кодера) или добавлен к "AttributeList" включающего элемента со значением "false" или "0". Опциональный компонент должен быть закодирован, опуская связанные теги.

33 Команда кодирования USE-NUMBER

33.1 Общие положения

33.1.1 "UseNumberInstruction":

UseNumberInstruction::=
    USE-NUMBER
    TargetList

33.1.2 Продукционное правило "TargetList" определено в 13.2.

33.1.3 Назначение этой команды кодирования состоит в том, чтобы модифицировать результат кодирования перечислимого типа так, чтобы числа в перечислениях "NamedNumber" использовались вместо имен.

33.2 Ограничения

33.2.1 Эта команда кодирования должна игнорироваться, если только она не применяется к перечислимому типу.

33.2.2 Тип с этой окончательной командой кодирования не должен также иметь окончательной команды кодирования TEXT.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не должны появляться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, ANY-ELEMENT, BASE64, DECIMAL, EMBED-VALUES, LIST, USE-NIL, USEORDER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE

33.2.3 В "TargetList" не должно быть уточняющей информации.

33.3 Действие на результат кодирования

33.3.1 "ExtendedXMLEnumeratedValue":

ExtendedXMLEnumeratedValue::=
    EmptyElementEnumerated
    | TextEnumerated
    | XMLSignedNumber

33.3.2 "EmptyElementEnumerated" и "TextEnumerated" определены в ИСО/МЭК 8824-1, 19.8 и 19.9.

33.3.3 "XMLSignedNumber" определено в ИСО/МЭК 8824-1, 18.9 и 18.12, и оно должно быть числом в перечислении "NamedNumber".

33.3.4 Альтернатива "XMLSignedNumber" должна использоваться, только и если только перечислимый тип имеет эту окончательную команду кодирования.

Примечание — Если GLOBAL-DEFAULTS MODIFIED-ENCODINGS присутствует в разделе управления кодированием XER, но перечислимый тип не имеет этой окончательной команды кодирования, то используется вторая альтернатива. Если в разделе управления кодированием XER нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS, то используется первая альтернатива.

34 Команда кодирования USE-ORDER

34.1 Общие положения

34.1.1 "UseOrderInstruction":

UseOrderInstruction::=
    USE-ORDER
    TargetList

34.1.2 Продукционное правило "TargetList" определено в 13.2.

34.1.3 Назначение этой команды кодирования состоит в том, чтобы позволить оптимизировать результат EXTENDED-XER кодирования типа последовательность, в котором есть компонент последовательность-из, определяющий семантический порядок значений следующих компонентов типа последовательность, которые кодируются как элементы. Это может также использоваться, если есть окончательная команда кодирования USE-NIL (см. раздел 32) и единичный компонент OPTIONAL, необходимый для использования USE-NIL в последовательности с целью определения семантического порядка компонентов той последовательности OPTIONAL.

34.1.4 Компонент последовательности-из, который определяет семантический порядок, это — первый компонент последовательности, если только нет также компонента последовательности-из, поддерживающего окончательную команду кодирования EMBED-VALUES, применяемую к типу

последовательность. В этом случае компонент последовательности-из, поддерживающий команду кодирования EMBED-VALUES, предшествует компоненту последовательности-из, поддерживающему команду кодирования USE-ORDER.

34.1.5 Компонент, определяющий семантический порядок, обязан быть типа последовательность-из с компонентом перечислимого типа. Этот тип последовательности-из и его семантика зависят от присутствия или отсутствия команды кодирования USE-NIL, применяемой к типу последовательность, как описано в следующих подпунктах.

34.1.5.1 Если нет окончательной команды кодирования USE-NIL, имена перечислений тождественны идентификаторам компонентов ACH.1 типа последовательность. Порядок перечислений в каждом абстрактном значении определяет семантический порядок значений последующих компонентов типа последовательность, которые присутствуют в результате кодирования.

34.1.5.2 Если есть окончательная команда кодирования USE-NIL, то OPTIONAL компонент, необходимый при использовании USE-NIL, обязан быть типа последовательность (скажем, B), и имена перечислений тождественны идентификаторам компонентов ACH.1 типа B последовательность. Порядок перечислений в каждой абстрактном значении определяет семантический порядок значений компонентов типа B последовательность, которые присутствуют в результате кодирования.

34.2 Ограничения

34.2.1 Эта команда кодирования должна быть назначена только типу последовательность. Тип последовательность должен содержать компонент типа последовательность-из (скажем, тип A) с компонентом перечислимого типа. Если тип последовательность не имеет также окончательных команд кодирования EMBED-VALUES, тогда тип A должен быть первым компонентом, а в противном случае — вторым. Если нет окончательной команды кодирования USE-NIL, тип последовательность должен также иметь хотя бы один компонент без окончательных команд кодирования ATTRIBUTE или ANY-ATTRIBUTES (компонент, не являющийся атрибутом). Если есть окончательная команда кодирования USE-NIL, то OPTIONAL компонент, поддерживающий USE-NIL, будет типа последовательность, и должен иметь, по крайней мере, один компонент.

34.2.2 Перечислимый тип должен иметь идентификаторы, которые зависят от присутствия или отсутствия окончательной команды кодирования USE-NIL, применяемой к типу последовательность с командой кодирования USE-ORDER, как определено в следующих подпунктах.

34.2.2.1 Если нет окончательной команды кодирования USE-NIL, то перечислимый тип должен иметь идентификаторы для перечисления, которые находятся во взаимно-однозначном соответствии (и находятся в том же текстовом порядке) с идентификаторами следующих компонентов последовательности, не являющихся атрибутами (см. 34.2.1). Тип последовательность-из должен быть ограничен так, чтобы каждое абстрактное значение содержало точно один идентификатор для каждого компонента последовательности, не являющегося атрибутом, который присутствует в абстрактном значении.

34.2.2.2 Если есть окончательная команда кодирования USE-NIL, то перечислимый тип должен иметь идентификаторы для перечислений, которые находятся во взаимно-однозначном соответствии (и находятся в том же текстовом порядке) с идентификаторами компонентов компонента OPTIONAL в типе последовательность. Тип последовательность-из должен быть ограничен так, чтобы каждое абстрактное значение содержало точно один идентификатор для каждого OPTIONAL компонента последовательности, который присутствует в абстрактном значении.

Примечание — Рекомендуется ограничение на тип последовательность выражать как:
(CONSTRAINED BY {/* Должен соответствовать ИСО/МЭК 8825-4, часть 35 */})

34.2.2.3 "EnumerationItem"s в перечислениях все должны быть «Идентификаторами» или «ПоименованнымиЧислами» ("NamedNumber") со значения 0 для первой позиции "EnumerationItem", 1 — для второй, и так далее, до последней позиции "EnumerationItem".

34.2.3 Тип последовательность-из не должен быть отмечен OPTIONAL или DEFAULT.

34.2.4 Следующие компоненты последовательности (если нет окончательной команды кодирования USE-NIL) и компонент OPTIONAL последовательности (если есть окончательная команда кодирования USE-NIL) не должны быть отмечены DEFAULT, если только они не имеют окончательной команды кодирования ATTRIBUTE.

34.2.5 Никакой компонент любой последовательности с этой окончательной командой кодирования или OPTIONAL последовательности (когда есть окончательная команда кодирования USE-NIL), не должен иметь окончательной команды кодирования UNTAGGED, независимо от того, является ли тип этого компонента посимвольно кодируемым или нет.

34.2.6 Никакой компонент последовательности с этой окончательной командой кодирования не должен иметь окончательной команды кодирования ANY-ELEMENT.

34.2.7 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

34.2.8 Тип с этой окончательной командой кодирования не должен иметь окончательной команды кодирования UNTAGGED.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не могут появляться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, ANY-ELEMENT, ATTRIBUTE, BASE64, DECIMAL, LIST, TEXT, USENUMBER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE.

34.2.9 В "TargetList" не должно быть уточняющей информации.

34.3 Действие на результат кодирования

34.3.1 Тип последовательность-из с перечислимым компонентом не должен кодироваться непосредственно.

34.3.2 Кодер должен закодировать семантику этого типа (семантический порядок компонентов последовательности или компонентов последовательности OPTIONAL), кодируя компоненты, которые кодируются как элементы в порядке, определенном типом последовательность-из с перечислимым компонентом. Декодер должен возвратить значение компонента последовательность-из, используя порядок кодируемых элементов.

35 Команда кодирования USE-QNAME

35.1 Общие положения

35.1.1 "UseQNameInstruction":

UseQNameInstruction::=

USE-QNAME

TargetList

35.1.2 Продукционное правило "TargetList" определено в 13.2.

35.1.3 Назначение этой команды кодирования состоит в том, чтобы модифицировать результат кодирования типа последовательность, каждое из значений которого определяет опциональное имя пространства имен (URI) и имя, не снабженное префиксом, так, что оно кодируется как имя XML, ограниченное пространством имен, или как неограниченное имя.

Примечание — Это обуславливается, поскольку такой вариант является доступным в других нотациях схемы. Пример типа последовательность, к которому это может быть применено — тип QName, определенный в ИСО/МЭК 8825-5.

35.1.4 Если опциональный компонент присутствует в абстрактном значении типа последовательность, то это абстрактное значение представляет имя, ограниченное пространством имен. Если дополнительный компонент отсутствует, тип последовательности представляет неограниченное имя.

35.2 Ограничения

35.2.1 Эта команда кодирования должна быть назначена последовательности с точно двумя компонентами, оба должны быть типа UTF8String. Первый компонент должен быть OPTIONAL.

35.2.2 Первый компонент должен быть ограничен, чтобы представить URI (см. IETF RFC 2396). Второй компонент должен быть ограничен, чтобы содержать "NCName", как определено в W3C XML Namespaces, разделе 2, продукционное правило 4, и не начинаться с символов, которые в верхнем регистре представляют собой "XML".

35.2.3 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

35.2.4 Тип с этой окончательной командой кодирования не должен иметь окончательной команды кодирования USE-NIL.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не должны появляться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, ANY-ELEMENT, BASE64, DECIMAL, EMBED-VALUES, LIST, TEXT, USENUMBER, USE-ORDER, USE-TYPE, USE-UNION, WHITESPACE.

35.2.5 В "TargetList" не должно быть никакой уточняющей информации.

35.3 Действие на результат кодирования

35.3.1 Эта команда кодирования, примененная к типу, если опциональный компонент присутствует, требует, чтобы декларация пространства имен (или декларация пространства имен по умолчанию) находилась в области применения для атрибута значения или контента элемента, который кодирует значение этого типа в соответствии с разделом 28. Значение атрибута или контент элемента тогда кодируются, как определено для имени, ограниченного пространством имен в разделе 28.

35.3.2 Если опциональный компонент отсутствует, то декларация пространства имен по умолчанию не должна быть в области применения для атрибута значения или контента элемента, который кодирует значение этого типа.

36 Команда кодирования USE-TYPE

36.1 Общие положения

36.1.1 "UseTypeInstruction":

UseTypeInstruction::=

USE-TYPE

TargetList

36.1.2 Продукционное правило "TargetList" определено в 13.2.

36.1.3 Эта команда кодирования оптимизирует EXTENDED-XER кодирование типа выбора. Атрибут идентификации типа должен быть закодирован в включающем элементе, чтобы идентифицировать альтернативу, которая была закодирована (если только это — не первая альтернатива), и удаление начального и конечного тегов результата кодирования альтернатив.

36.1.4 Атрибут идентификации типа идентифицирует тип XML элемента. Необходимо, чтобы имя атрибута было именем "типа" ("type") из управляющего пространства имен (см. 15.9) и его значение идентифицировало бы альтернативу типа выбора, к которому эта команда кодирования применена (это обеспечивает альтернативное определение для типа выбора).

36.2 Ограничения

36.2.1 Тип, которому назначена USE-TYPE, должен быть типом выбора без окончательной команды кодирования UNTAGGED.

36.2.2 Ни одна из альтернатив для типа выбора не должна иметь окончательную команду кодирования UNTAGGED.

36.2.3 Ни одна из альтернатив типа выбора не должна быть сама типом выбора с окончательной командой кодирования USE-TYPE.

Примечание — Одна или более альтернатив для типа выбора могут быть типами выбора с окончательной командой кодирования USE-UNION, кодирующим.

36.2.4 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

36.2.5 Тип с этой окончательной командой кодирования не должен иметь ни одной из окончательных команд кодирования UNTAGGED или USE-UNION.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не должны появляться с этой окончательной командой кодирования, потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, ANY-ELEMENT, ATTRIBUTE, BASE64, DECIMAL, DEFAULT-FOR-EMPTY, EMBED-VALUES, LIST, TEXT, USE-NIL, USE-NUMBER, USE-ORDER, USE-QNAME, WHITESPACE.

36.2.6 В "TargetList" не должно быть уточняющей информации.

36.3 Действие на результат кодирования

36.3.1 Если кодируемая альтернатива выбора не является первой альтернативой этого выбора, то атрибут идентификации типа (см. 36.3.3 и 36.3.4) должен быть добавлен к "AttributeList" включающего элемента, если 36.3.8 не применяется.

36.3.2 Если кодируемая альтернатива выбора является первой альтернативой этого выбора, атрибут идентификации типа может быть добавлен или опущен как опция кодера, если 36.3.8 не применяется.

36.3.3 Атрибут идентификации типа должен быть экземпляром продукционного правила "Attribute" (см. 19.3.3) с "ControlAttributeName" пространства имен ограниченного использования (см. 19.3.5) "типа" ("type") из управляющего пространства имен (см. 15.9).

36.3.4 Значение атрибута идентификации типа должно быть идентификатором выбранной альтернативы, возможно модифицированной в соответствии с любой окончательной командой кодирования NAME или NAMESPACE.

36.3.5 Если нет атрибута идентификации типа, представленного в результате кодирования типа с этой окончательной командой кодирования, декодер должен предположить, что представлена первая альтернатива выбора.

36.3.6 Присутствие атрибута идентификации типа с непредвиденным значением не должно приводить к ошибке декодирования. Сталкиваясь с таким атрибутом в результате кодирования, декодер должен предположить, что была идентифицирована первая альтернатива выбора, и может проигнорировать атрибут идентификации типа (или передать его приложению). Кроме того, в таких случаях декодер может проигнорировать (или передать приложению) любые другие непредвиденные атрибуты и любые непредвиденные дочерние элементы, встречающиеся после всех ожидаемых дочерних элементов в альтернативе "ExtendedXMLValue".

36.3.7 Все "Атрибуты" ("Attribute"s), которые были бы иначе в "AttributeList" "ExtendedXMLChoiceValue", должны быть добавлены к "AttributeList" включающего элемента и "ExtendedXMLChoiceValue" типа выбора должно быть заменено "ExtendedXMLValue" в "ExtendedXMLChoiceValue".

36.3.8 Если одна или более альтернатив типа выбора с окончательной командой кодирования USE-TYPE относятся к типу выбора с окончательной командой кодирования USE-UNION, атрибут идентификации типа как может опция кодера идентифицировать одну из альтернатив типа выбора с окончательной командой кодирования USE-UNION вместо альтернативы типа выбора с окончательной командой кодирования USE-TYPE.

37 Команда кодирования USE-UNION

37.1 Общие положения

37.1.1 "UseUnionInstruction":

UseUnionInstruction::=

USE-UNION

TargetList

37.1.2 Продукционное правило "TargetList" определено в 13.2.

37.1.3 Эта команда кодирования оптимизирует результат кодирования типа выбор в тех случаях, когда результат кодирования абстрактных значений каждой альтернативы существенно отличается от результата кодирования абстрактных значений других альтернатив для декодера, определяющего абстрактные значения, представленные анализом кодирования.

37.1.4 Если тип выбора с окончательной командой кодирования USE-UNION не имеет окончательных команд кодирования ATTRIBUTE или UNTAGGED, тогда эта команда кодирования может привести к вставке атрибута идентификации типа в включающий элемент, чтобы идентифицировать альтернативу, которая была закодирована. Если тип выбора имеет окончательную команду кодирования ATTRIBUTE или UNTAGGED, или является компонентом типа последовательность-из или набор-из type командой кодирования LIST, вставка атрибута идентификации типа невозможна.

37.1.5 Эта команда кодирования вызывает удаление начального и конечного тегов результата кодирования альтернативы.

37.2 Ограничения

37.2.1 Тип с окончательной командой кодирования USE-UNION должен быть типом выбора.

37.2.2 Все альтернативы типа выбора должны быть посимвольно кодируемыми типами, но не должны быть типами выбора с окончательной командой кодирования USE-UNION.

37.2.3 Если тип выбора имеет окончательную команду кодирования ATTRIBUTE или UNTAGGED или используется в определении типа как компонент типа последовательность-из или набор-из с окончательной командой кодирования LIST, альтернативы типа выбора должны быть ограничены так, чтобы для любой альтернативы все ее абстрактные значения имели, по крайней мере, один результат кодирования (ее "ExtendedXMLValue"), который отличается от всех разрешенных результатов кодирования всех предшествующих по тексту альтернатив.

Примечание — Это требование обусловлено невозможностью вставить атрибут идентификации типа, определяющий альтернативу, которая была выбрана. Без этого требования кодирование не было бы однозначным.

37.2.4 В следующих двух подразделах термин "identifier" означает: идентификатор (возможно модифицированый в соответствии с любой окончательной командой кодирования NAME and NAMESPACE) альтернативы (типа выбора).

37.2.5 Если тип выбора (скажем, тип U), кодируется как альтернатива включающего типа выбора (скажем, типа E), с окончательной командой кодирования USE-TYPE и идентификатор одной из альтернатив E тождествен идентификатору одной из альтернатив U, тогда каждое абстрактное значение этой альтернативы U должно иметь по крайней мере один результат кодирования, который отличается от всех результатов кодирования предшествующих по тексту альтернатив U.

Примечание — Это требование обусловлено невозможностью идентифицировать альтернативу U, так как идентификатор в атрибуте идентификации типа для E просто идентифицировал бы весь U.

37.2.6 Если тип выбора (скажем, U1) кодируется как альтернатива включающего типа выбора (скажем, типа E) с окончательной командой кодирования USE-TYPE и E содержит другой тип выбора (скажем, U2) с командой кодирования USE-UNION, которая следует по тексту за U1 в E, и идентификатор любой из альтернатив U2 тождествен одному из идентификаторов U1, тогда каждое абстрактное значение этой альтернативы U2 должно иметь, по крайней мере, один результат кодирования, который отличается от всех результатов кодирования предшествующих по тексту альтернатив U1.

Примечание — Это требование обусловлено невозможностью идентифицировать альтернативу U2, так как идентификатор в атрибуте идентификации типа для E идентифицировал бы альтернативу в U1.

37.2.7 Эта команда кодирования не должна быть назначена, если нет команды кодирования GLOBAL-DEFAULTS MODIFIED-ENCODINGS в разделе управления кодированием.

37.2.8 Тип с этой окончательной командой кодирования не должен также иметь окончательной командой кодирования USE-TYPE.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не могут появляться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, ANY-ELEMENT, BASE64, DECIMAL, EMBED-VALUES, LIST, TEXT, USENIL, USE-NUMBER, USE-ORDER, USE-QNAME, WHITESPACE.

37.2.9 В "TargetList" не должно быть уточняющей информации.

37.3 Действие на результат кодирования

37.3.1 Если тип выбора не имеет окончательной командой кодирования ATTRIBUTE или UNTAGGED и не закодирован как компонент типа последовательность-из или набор-из с окончательной командой кодирования LIST, тогда атрибут идентификации типа может быть добавлен как опция кодера к "AttributeList" включающего элемента (но см. 37.3.8).

Примечание — Если тип выбора закодирован как альтернатива выбора с командой кодирования USE-TYPE, атрибут идентификации типа, определенный командой кодирования USE-UNION, может использоваться вместо атрибута идентификации типа, определенного командой кодирования USE-TYPE (см. 36.3.8).

37.3.2 Если каждый возможный результат кодирования кодируемого абстрактного значения идентичен одному из результатов кодирования абстрактного значения предшествующей по тексту альтернативы, то должен быть добавлен атрибут идентификации типа.

Примечание — Подраздел 37.3.2 отменяет опцию кодера подраздела 37.3.1 и делает дополнение атрибута идентификации типа обязательным. Ограничения, определенные в 37.2.4 к 37.2.6 гарантируют, что это может произойти, только когда тип выбора закодирован как элемент и когда никакая двусмысленность из-за идентичных идентификаторов невозможна.

37.3.3 Если тип выбора имеет окончательную команду кодирования ATTRIBUTE или UNTAGGED, или его включающий тип является типом последовательность-из или набор-из с окончательной командой кодирования LIST, никакой атрибут идентификации типа не может быть вставлен ни в один элемент. В случае сценариев, описанных в 37.2.4 — 37.2.6, атрибут идентификации типа не может быть вставлен для точной идентификации некоторых альтернатив U или U2. Декодеры должны поэтому полагаться на условия 37.2.4 — 37.2.6 для определения абстрактного значения, которое было закодировано.

Примечание — Эти правила подразумевают, что декодер необходим в отсутствие идентификации типа (или при наличии неоднозначной идентификации), чтобы попытаться декодировать первую по тексту альтернативу, затем следующую и так далее, принимая первое успешное декодирование (или диагностирование ошибки, если нет ни одного успешного декодирования).

37.3.4 Атрибут идентификации типа должен быть экземпляром продукционного правила "Attribute" (см. 19.3.3) с "ControlAttributeName" пространства имен ограниченного использования (см. 19.3.5) «типа» ("type") из управляющего пространства имен (см. 15.9).

37.3.5 Значение атрибута идентификации типа должно быть идентификатором выбранной альтернативы, возможно, модифицированной в соответствии с любой окончательной командой кодирования NAME и NAMESPACE.

37.3.6 Все "Атрибуты" ("Attribute"s), которые были бы иначе в "AttributeList" "ExtendedXMLChoiceValue", должны быть добавлены к "AttributeList" включающего элемента, и "ExtendedXMLChoiceValue" типа выбора должно быть заменено "ExtendedXMLValue" в "ExtendedXMLChoiceValue".

37.3.7 "ExtendedXMLValue" посимвольно кодируемого типа должно быть одним из результатов кодирования, которое не содержит любые XML теги.

Примечание — Это может ограничить опции кодера.

37.3.8 Если альтернатива типа выбора имеет окончательную команду кодирования NAME AS " ", никакой атрибут идентификации типа не должен быть добавлен для этой альтернативы.

38 Команда кодирования WHITESPACE

38.1 Общие положения

38.1.1 "WhiteSpaceInstruction":

```
WhiteSpaceInstruction::=
        WHITESPACE
        TargetList
        WhiteSpaceAction
WhiteSpaceAction::=
        REPLACE
        | COLLAPSE
```

38.1.2 Продукционное правило "TargetList" определено в 13.2

38.1.3 Эта команда кодирования требует, чтобы декодеры приняли дополнительные опции при кодировании символа SPACE(32) и при использовании начальных и замыкающих «пробелов с переходом» (см. 7.1.5) для кодирования строки символов.

38.2 Ограничения

38.2.1 Эта команда кодирования может быть назначена только типу ограниченной строки символов, которая не содержит или вынуждена не содержать следующие символы:

а) ГОРИЗОНТАЛЬНАЯ ТАБУЛЯЦИЯ (9);

б) ПЕРЕВОД СТРОКИ (10);

в) ПЕРЕВОД КАРЕТКИ (13).

38.2.2 Если команда кодирования имеет опцию COLLAPSE, то она не должна применяться к типу ограниченной строки символов, если только этому типу не запрещено иметь ввод и отслеживание пробела или содержать множественные смежные пробелы для любого абстрактного значения.

Примечание — Признано, что некоторые инструменты АСН.1 не могут статически проверять, что вышеуказанное ограничение будет удовлетворяться для всех абстрактных значений, но соответствующие кодеры не могут генерировать результаты кодирования, в которых "ExtendedXMLValue" нарушает это ограничение.

38.2.3 Тип с этой окончательной командой кодирования не должен также иметь любую из двух окончательных команд кодирования ANY-ELEMENT или BASE64.

Примечание — Нижеперечисленные окончательные команды кодирования никогда не могут появляться с этой окончательной командой кодирования потому, что их применение к этому типу запрещено: ANY-ATTRIBUTES, DECIMAL, EMBED-VALUES, LIST, TEXT, USE-NIL, USE-NUMBER, USE-ORDER, USE-QNAME, USE-TYPE, USE-UNION.

38.2.4 В "TargetList" не должно быть уточняющей информации.

38.3 Действие на результат кодирования

38.3.1 Если используется ключевое слово REPLACE, каждый SPACE(32) может быть заменен в качестве опции кодера единственным символом "пробел с переходом" (см. 7.1.5).

38.3.2 Если используется ключевое слово COLLAPSE, каждый SPACE(32) может быть заменен в качестве опции кодера любым количеством символов «пробел с переходом». Кроме того, один или более таких символов могут быть добавлены к началу или к концу результата кодирования “ExtendedXMLValue” в качестве опции кодера.

39 Значения идентификатора объекта, ссылающиеся на правила кодирования

39.1 Правила кодирования, определенные в этом Международном стандарте могут использоваться для ссылки и применяться всегда, когда есть необходимость определить однозначное представление строки символов для значений единственного идентифицированного типа ACH.1.

39.2 Следующие значения идентификатора объекта и дескриптора объекта назначаются, чтобы идентифицировать правила кодирования, определенные в настоящем стандарте:

Для BASIC-XER:

{joint-iso-itu-t ACH1 (1) xml-encoding (5) basic (0)}
"Basic XML encoding of a single ACH.1 type"

Для CXER:

{joint-iso-itu-t ACH1 (1) xml-encoding (5) canonical (1)}
"Canonical XML encoding of a single ACH.1 type"

Для EXTENDED-XER:

{joint-iso-itu-t ACH1 (1) xml-encoding (5) extended (2)}
"Extended XML encoding of a single ACH.1 type"

39.3 Следующие значения идентификатора объекта и дескриптора объекта назначаются, чтобы идентифицировать пространство имен ACH.1 (см. 15.9):

ACH1Namespace OBJECT IDENTIFIER::=
{joint-iso-itu-t ACH1 (1) xml-encoding (5) extended (2)
modules (0) support (1)}
"ACH.1 namespace for EXTENDED-XER support"

Приложение А
(справочное)

Примеры кодирования BASIC-XER и CXER

Данное приложение иллюстрирует применение базовых и канонических правил XML кодирования, определенных в настоящем стандарте. Приложение содержит XML разметку представления (гипотетической) записи данных о персонале, которая определена с использованием АСН.1.

А.1 АСН.1 описание структуры записи

Структура гипотетической кадровой записи данных описана ниже с использованием АСН.1, определенной в ИСО/МЭК 8824-1. Приведенный пример идентичен описанному в приложении А ИСО/МЭК 8825-1.

```
PersonnelRecord ::= [APPLICATION 0] IMPLICIT SET {
        name            Name,
        title           [0] VisibleString,
        number          EmployeeNumber,
        dateOfHire      [1] Date,
        nameOfSpouse    [2] Name,
        children        [3] IMPLICIT
    SEQUENCE OF ChildInformation DEFAULT {} }
ChildInformation ::= SET
        { name      Name,
        dateOfBirth [0] Date}
Name ::= [APPLICATION 1] IMPLICIT SEQUENCE
        { givenName     VisibleString,
         initial        VisibleString,
         familyName     VisibleString}
EmployeeNumber ::= [APPLICATION 2] IMPLICIT INTEGER
Date ::= [APPLICATION 3] IMPLICIT VisibleString – YYYYMMDD
```

Примечание — В этом примере используются теги только для сравнения с примером ранней версии ИСО/МЭК 8824-1. Они не влияют на BASIC-XER, CXER и EXTENDED-XER XML кодирование.

А.2 АСН.1 описание значений записи

Значение кадровой записи данных John Smith's описано ниже, с использованием основной нотации значений АСН.1:

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

А.3 Базовое XML представление значения записи

Представление значения записи, приведенное в А.2 (после применения базовых правил XML кодирования, определенных в настоящем стандарте), показано ниже, предполагается пустой пролог.

Длина этого кодирования в BASIC-XER — 653 октета, без всех разделителей. Для сравнения та же запись, кодированная с UNALIGNED вариантом PER (см. ИСО/МЭК 8825-1) — 84 октета, с ALIGNED вариантом PER — 94 октета, с BER (см. ИСО/МЭК 8825-2) с использованием формы определенной длины — минимум 136 октетов, и с BER с использованием формы неопределенной длины — минимум 161 октет.

```
<PersonnelRecord>
    <name>
        <givenName>John</givenName>
        <initial>P</initial>
        <familyName>Smith</familyName>
    </name>
    <title>Director</title>
    <number>51</number>
    <dateOfHire>19710917</dateOfHire>
    <nameOfSpouse>
```

```
            <givenName>Mary</givenName>
            <initial>T</initial>
            <familyName>Smith</familyName>
        </nameOfSpouse>
        <children>
            <ChildInformation>
                <name>
                    <givenName>Ralph</givenName>
                    <initial>T</initial>
                    <familyName>Smith</familyName>
                </name>
                <dateOfBirth>19571111</dateOfBirth>
            </ChildInformation>
            <ChildInformation>
                <name>
                    <givenName>Susan</givenName>
                    <initial>B</initial>
                    <familyName>Jones</familyName>
                </name>
                <dateOfBirth>19590717</dateOfBirth>
            </ChildInformation>
        </children>
</PersonnelRecord>
```

А.4 Каноническое XML представление значения записи

Представление значения записи, приведенное в А.3 (после применения канонических правил XML кодирования, определенных в настоящем стандарте), показано ниже:

```
<PersonnelRecord><name><givenName>John</givenName><initial>P</initial><familyName>Smith</fam
ilyName></name><number>51</number><title>Director</title><dateOfHire>19710
917</dateOfHire><nameOfSpouse><givenName>Mary</givenName><initial>T</initial><familyName>Smith</
familyName></name
OfSpouse><children><ChildInformation><name><givenName>Ralph</givenName><initial>T</initia
l><familyName>Smith</familyName></name><dateOfBirth>19571111</dateOfBirth></ChildInformation
><ChildInformation><name><givenName>Susan</givenName><initial>B</initial><familyName>Jones</
familyName></name><dateOfBirth>19590717</dateOfBirth></ChildInformation></children></Personn
elRecord>
```

Приложение Б
(справочное)

Частичное XML содержание и детерминированные кодирования

Б.1 Частичное XML содержание

Примечание — Это приложение описывает действия, когда используется MODIFIED-ENCODINGS.

Б.1.1 В следующих подпунктах описывается создание частичного содержания XML элемента. Частичное содержание XML элемента производится в качестве части кодировок, и Б.2 определяет ограничения на частичное содержание XML элемента, которые необходимы для удовлетворения требований 9.2.11. Если спецификация ACH.1 с кодировкой XER инструкций не нарушает эти ограничения, она является допустимой спецификацией и ее допустимость легко проверить. В случае нарушения ограничений спецификация может не нарушать нормативных требований 9.2.11, но проверить это трудно.

Примечание — Ограничения направлены на обеспечение того, чтобы декодер мог легко и однозначно восстановить абстрактные значения, которые были использованы кодером при кодировании.

Б.1.2 Частичное содержание XML элемента состоит из комбинации отдельных XML элементов, предоставляемых [ELEMENT] SEQUENCE, SET, SEQUENCE OF, SET OF или CHOICE, и другими частичными содержаниями XML элемента, предоставляемых [UNTAGGED] SEQUENCE, SET, SEQUENCE OF, SET OF или CHOICE.

Примечание — Граница между частичным содержанием XML элемента в рамках более крупного частичного содержания XML элемента не видна в той кодировке, но может быть определена из схемы ACH.1 и ограничений на имена элементов.

Б.1.3 Возможные варианты частичного содержания XML элемента:

а) единичный элемент XML;

б) группа конкатенации, состоящая из упорядоченной конкатенации нуля, одного или более частичных содержаний XML элементов, в которой некоторое частичное содержание XML элемента может отсутствовать в момент кодировки (представляя отсутствие опционального абстрактного значения);

Примечание — Кодирование типа [UNTAGGED] SEQUENCE или SET будет в общем случае продуцировать группу конкатенации.

в) группа повторений, состоящая из повторения (неограниченного или ограниченного) содержания частичного XML элемента (называемого повторяемый компонент) создаваемого из компонентов SEQUENCE OF или SET OF;

Примечание — Кодирование типа [UNTAGGED] SEQUENCE OF или SET OF будет в целом производить группу повторений.

г) группа альтернатив, включающая одно содержание частичного XML элемента, выбранного из множества альтернативных содержаний частичного XML элемента (из которых точно один присутствует в кодировании).

Примечание — Кодирование типа CHOICE производит альтернативы группы. Каждый альтернативный тип CHOICE производит один из альтернативных содержаний частичного XML элемента для этого типа.

Б.2 Рекомендуемые ограничения на кодирования, производящие содержание частичного XML элемента

Б.2.1 Только для целей настоящего пункта любые группы повторения рассматриваются как если бы они были факультативными, т.е. могли бы иметь нулевые повторения.

Примечание — Это ограничение не является строго необходимым, если существуют ограничения, которые требуют, по крайней мере, одного повторения соответствующего типа ACH.1.

Б.2.2 Только для целей настоящего пункта требование различия имен элементов можно интерпретировать следующим образом:

а) все сравнения сделаны после применения каких-либо окончательных команд кодирования NAME и NAMESPACE по типу, который генерирует имя;

б) имена уточненного пространства имен отличны от неквалифицированных имен;

в) имена уточненного пространства имен отличаются, если и только если отличаются их unprefixed имена, имена пространства имен или оба варианта одновременно.

Б.2.3 Для любого получающегося содержания частичного XML элемента есть возможные разночтения (и, следовательно, возможность нарушения 9.2.11) если условия, указанные в настоящем подпункте, не выполня-

ются для всех возможных вариантов альтернатив в группе альтернатив, для всех возможных осуществлений выборочности в группе конкатенаций, для всех возможных повторений в группе повторений, и для всех возможных упорядочений кодировки компонентов набора.

Примечание — При чтении и реализации следующих пунктов текст уточнения «для всех возможных» очень важен. Создатели инструментов, определяющих, что является однозначной спецификацией, а что нет, придется анализировать все возможные комбинации вариантов, выборочности, повторений и порядков.

Б.2.4 Определение требования: не должно быть двух соседних содержаний частичных XML элементов с одинаковым именем элемента для первого элемента содержания второго частичного XML элемента и содержание последнего элемент содержание первого частичного XML элемента, если первый частичный XML элемента не требует саморазграничения.

Пример 1 — Содержание частичного XML элемента созданного [UNTAGGED] SEQUENCE, саморазграничено, если оно не заканчивается элементом OPTIONAL.

Пример 2 — Содержание частичного XML элемента созданного [UNTAGGED] SEQUENCE OF, саморазграничено, если оно имеет фиксированное число итераций, которые саморазграниченными. Это означает, в частности, что SEQUENCE OF [UNTAGGED] SEQUENCE OF INTEGER является неоднозначным и нарушает 9.2.11, если число повторений во второй SEQUENCE OF не будет фиксированным.

Пример 3 — Содержание частичного XML элемента, созданного [UNTAGGED] SET, никогда не саморазграничивается, если оно имеет любые дополнительные элементы.

Б.2.5 Альтернативное определение требований: первые XML элементы альтернативного содержания частичного XML элемента в группе альтернатив должны иметь различные имена элементов.

Примечание — Приведенный выше текст не учитывает возможного использования USE-TYPE и USE-UNION, которые выходят за рамки настоящего приложения.

Пример 4 — Кодировка:

```
BadExample1::= CHOICE {
    - - Первая альтернатива содержания частичное XML элемента
    alt1 [UNTAGGED] SEQUENCE {
        name UTF8String,
        zip-code UTF8String },
    alt2 [UNTAGGED] SEQUENCE {
        name UTF8String,
        post-code UTF8String } }
```

на самом деле не неоднозначная EXTENDED-XER кодировка (для декодера человека), но она нарушает вышеуказанные требования, а также нарушает 9.2.11. Это неправильное использование команды кодирования.

Б.2.6 Вспомогательное определение требований: имена XML элементов первого элемента XML всех последовательностей опционального содержания частичного XML элемента плюс, что в следующем/следующих обязательных содержаниях частичного XML элемента должны отличаться.

Примечание — Это означает, в частности, что любое опциональное содержание частичного XML элемента в конце группы, которая повторяется, и опциональное содержание частичного XML элемента в ее начале должны иметь различные имена XML элемента, если количество повторений ограничено максимум 1. Если все содержание частичного XML элемента группы, которая повторяется, не является обязательным, то их имена XML элементов должны быть различны.

Пример 5 — Кодировка:

```
BadExample2::= SEQUENCE OF {
    [UNTAGGED] SEQUENCE {
        first       [UNTAGGED] CommonInitialParms,
        second      MainInformation,
        third       [UNTAGGED] CommonEndParms } }
где
CommonInitialParms::= SEQUENCE { date GeneralizedTime OPTIONAL,
                                 married BOOLEAN}
CommonEndParms::= SEQUENCE { name UTF8String,
                             date GeneralizedTime OPTIONAL}
```

нарушает требования опционального определения и нарушает 9.2.11. Это неправильное использование команды кодирования.

Б.2.7 Требование определения числа повторений: все группы повторения, имеющие число повторений, которое не закреплено, должны сопровождаться содержанием частичного XML элемента, чей первый XML

элемент имеет имя, отличное от имени первого XML элемента содержания частичного XML элемента, которое повторяется.

П р и м е р 6 — Кодировка:

```
BadExample3::= SEQUENCE {
    required-items [UNTAGGED] SEQUENCE OF Book,
    optional-items [UNTAGGED] SEQUENCE OF Book }
```

нарушает требование определения числа повторений, а также нарушает 9.2.11. Это неправильное использование команды кодирования. Альтернатива:

```
GoodExample1::= SEQUENCE {
    required-items [UNTAGGED] SEQUENCE OF required-books Book ,
    optional-items [UNTAGGED] SEQUENCE OF optional-books Book }
```

является правильным использованием команды кодирования.

Б.2.8 Требование определения набора компонентов: первый XML элемент в содержании частичного XML компонента группы конкатенаций, который кодируется как тип набора, должен иметь имя XML элемента, которое отличается от имени первого XML элемента в содержании частичного XML всех других компонентов.

П р и м е р 7 — Кодировка:

```
BadExample4::= SET {
    uk-mailing UNTAGGED] SEQUENCE {name UTF8String, post-code UTF8String}
    us-mailing [UNTAGGED] SEQUENCE {name UTF8String, zip-code UTF8String}}
```

нарушает требования определения компонентов, а также нарушает 9.2.11. Это неправильное использование команды кодирования.

Альтернатива:

```
GoodExample2 ::= SET {
    uk-mailing UNTAGGED] SEQUENCE {uk-name UTF8String, post-code UTF8String}
    us-mailing [UNTAGGED] SEQUENCE {us-name UTF8String, zip-code UTF8String}}
```

является правильным использованием команды кодирования.

Приложение В
(справочное)

Примеры EXTENDED-XER кодирования с использованием команд кодирования XER

В.1 Введение

В.1.1 Это приложение содержит обучающую информацию и примеры применения команд кодирования XER.

Примечание — Все примеры ACH.1 в этом приложении предполагают использование среды AUTOMATIC TAGS.

В.1.2 Обычно команды кодирования требуется назначать спецификации ACH.1 только в том случае, когда разработчику требуется соответствие фактической формы XML кодирования форме, определенной другими спецификациями или приемлемой для других инструментов XML. В противном случае, может быть использована только ACH.1 (с кодированием BASIC-XER или CXER).

В.1.3 Если ACH.1 используется как нотация определения схемы, то дополнительное использование команд кодирования будет в общем случае обеспечивать более компактное XML кодирование, чем использование только ACH.1, но результаты кодирования будут представлять собой гораздо более длинные тексты, чем при использовании ACH.1 с Правилами Уплотненного Кодирования (PER).

Примечание — Примеры (а также идентификаторы и использованные имена типов), предназначены для иллюстрации особенностей EXTENDED-XER и, в общем, не представляют собой реальных спецификаций.

В.1.4 Команды кодирования XER делятся на две большие категории.

В.1.5 В первую категорию попадают команды кодирования, которые, обычно, полезны при разработке формы документа XML. Они, как правило, допустимы даже при отсутствии DEFAULTS GLOBAL MODIFIED-ENCODINGS. Двумя наиболее полезными из них являются ATTRIBUTE и LIST, и в п В.2 представлены простые примеры их использования.

В.1.6 Во вторую категорию попадают команды кодирования, которые предназначены для поддержки отображения из W3C XML Schema, специфицированного в ИСО/МЭК 8825-5. Они обычно требуют наличия DEFAULTS GLOBAL MODIFIED-ENCODINGS в разделе управления кодированием, но это не показано в примерах. В этих примерах предполагается, что любой тип ссылки, начинающийся с "XSD.", будет импортирован из приложения А ИСО/МЭК 8825-5. В пункте В.3 приводятся примеры их использования. **Эти примеры не являются полными модулями ACH.1 и полными XML-документами:** модуль заголовков, как правило, опущен, а также предполагается, что любой XML атрибут, начинающийся с "asn1:", будет управляющим атрибутом, использующим пространство имен asn1 для управляющего атрибута, когда предполагается, что префикс "asn1" уже декларирован. (На практике, если источником результата кодирования является схема W3C XML Schema, более вероятно использование префикса "xsi" с пространством имен XSI).

В.1.7 Почти во всех случаях команды кодирования, снабженные префиксами, используются для ясности, хотя в реальной спецификации большая краткость (и более четкое отделение определения абстрактного синтаксиса от проблем кодировки) будет достигнута за счет использования Раздела Управления Кодированием.

В.2 Простые примеры

В.2.1 Бейсбольная карточка:

```
BBCard::= SEQUENCE {
        name [ATTRIBUTE] IA5String,
        team [ATTRIBUTE] IA5String,
        age INTEGER,
        position IA5String,
        handedness ENUMERATED {
        left-handed,
        right-handed,
        ambidextrous },
        batting-average REAL }
```

Игнорируя команду кодирования (BASIC-XER), мы получим:

```
<BBCard>
        <name>Jorge Posada</name>
        <team>New York Yankees</team>
        <age>29</age>
        <position>C</position>
        <handedness><right-handed/></handedness>
```

```
        <batting-average>0.277</batting-average>
    </BBCard>
```

Результат кодирования EXTENDED-XER (с MODIFIED-ENCODINGS) того же значения будет:

```
    <BBCard name = Jorge Posada team = New York Yankees>
        <age>29</age>
        <position>C</position>
        <handedness>right-handed</handedness>
        <batting-average>0.277</batting-average>
    </BBCard>
```

B.2.2 Сотрудник

```
Employee::= [NAME AS UNCAPITALIZED] SEQUENCE {
    id [ATTRIBUTE] INTEGER(0..MAX),
    recruited XSD.Date,
    salaries [LIST] SEQUENCE
    OF salary REAL }
```

Игнорируя команду кодирования (BASIC-XER), мы получим:

```
    <Employee>
        <id>239</id>
        <recruited>27-11-2002</recruited>
        <salaries>
            <salary>29876</salary>
            <salary>54375</salary>
            <salary>98435</salary>
        </salaries>
    </Employee>
```

Результат кодирования EXTENDED-XER того же же значения будет:

```
    <employee id = "239">
        <recruited>27-11-2002</recruited>
        <salaries>29876 54375 98435</salaries>
    </employee>
```

Используя Раздел Управления Кодированием XER, мы получим:

```
    Employee ::= SEQUENCE {
        id          INTEGER(0..MAX),
        recruited Date,
        salaries SEQUENCE OF salary REAL }
    ENCODING-CONTROL XER
        NAME Employee AS UNCAPITALIZED
            ATTRIBUTE Employee.id
            LIST  Employee.salaries
```

B.3 Более сложные примеры

B.3.1 Использование объединения двух простых типов

```
Int-or-boolean::= [USE-UNION] CHOICE {
    int INTEGER,
    boolean BOOLEAN }
```

Результаты кодирования будут:

```
    <Int-or-boolean><int>39</int></Int-or-boolean>   BASIC-XER
    <Int-or-boolean><boolean><true/></boolean></Int-or-boolean>   BASIC-XER
    <Int-or-boolean>39</Int-or-boolean>   EXTENDED-XER
    <Int-or-boolean>true</Int-or-boolean>   EXTENDED-XER
```

B.3.2 Использование атрибута идентификации типов

```
Int-or-boolean::= [USE-TYPE] CHOICE {
    int INTEGER,
    boolean BOOLEAN }
```

Результаты кодирования будут:

```
    <Int-or-boolean><int>39</int></Int-or-boolean>      BASIC-XER
    <Int-or-boolean><boolean><true/></boolean></Int-or-boolean>      BASIC-XER
    <Int-or-boolean asn1:type="int">39</Int-or-boolean>      EXTENDED-XER
    <Int-or-boolean asn1:type="boolean">true</Int-or-boolean>      EXTENDED-XER
```

B.3.3 Использование перечислимых значений

```
PrimesUnder30::= [USE-NUMBER] ENUMERATED {
        int2(2), int3(3), int5(5), int7(7), int11(11), int13(13), int17(17), int19(19), int23(23), int29(29)}
InputValues::= [ATTRIBUTE] [LIST] SEQUENCE OF PrimesUnder30
PrimeProducts::= SEQUENCE {
        input InputValues,
        output [ATTRIBUTE] [DECIMAL] REAL}
```

Результаты кодирования будут:

```
<PrimeProducts>
        <input><int2/><int7/><int17/><int23/><int29/><int3/></input>
        <output>476338.00</output>
</PrimeProducts>  BASIC-XER
<PrimeProducts input="2 7 17 23 29 3" output="476338.00"/>
  EXTENDED-XER
```

B.3.4 Использование пустого результата кодирования для значений по умолчанию

```
Responses::= ENUMERATED {ringing, engaged, number-not-known }
CallDetails::= [DEFAULT-FOR-EMPTY number-not-known] SEQUENCE {
        number [ATTRIBUTE] NumericString,
        response Response }
```

Результаты кодирования будут:

```
<CallDetails>
        <number>0164593746</number>
        <response><number-not-known/></response>
</CallDetails>  BASIC-XER
<CallDetails number="0164593746"/>  EXTENDED-XER
```

B.3.5 Использование встроенных значений для уведомления об обязательном платеже

```
Notification::= SEQUENCE {
        text [EMBED-VALUES] SEQUENCE OF UTF8String,
        account INTEGER,
        amount-due INTEGER,
        payable-by XSD:Date } (CONSTRAINED BY {/* Должно
        соответствовать ИСО/МЭК 8825-4, 25.2 */})
```

Значения в основой нотации значений АСН.1 будут:

```
firstNotification Notification::= {
        text {"Please note the following details:", "(your business account)",
            "This is in excess of your  normal monthly allowance", "or earlier"},
        account 568903,
        amount-due 536,
        payable-by "27-08-2003" }
```

Результат EXTENDED-XER кодирования будет иметь вид:

```
<Notification>
    Please note the following details:
    <account>568903</account>
    (your business account)
    <amount-due>536</amount-due>
    This is in excess of your normal monthly allowance
    <payable-by>27-08-2003</payable-by>
    or earlier
</Notification>
```

Приложение ГА
(справочное)

Сведения об учете в тексте стандарта Изменения 1. EXTENDED-XER

Таблица ГА.1

Раздел, подраздел, пункт, подпункт, таблица, приложение	Внесенные изменения
Введение	Добавлен текст: «BASIC-XER или CXER, то для XML представления применяется единственный фиксированный стиль без управления стилистическими особенностями пользователем. Настоящий стандарт специфицирует синтаксис и семантику команд XER кодирования, которые определяют стилистические особенности XML в EXTENDED-XER кодировании. Команды XER кодирования могут также быть использованы для определения возможного включения команд XML обработки в EXTENDED-XER кодирование. Команды XER кодирования игнорируются правилами BASIC-XER и CXER, но используются EXTENDED-XER. Примечание — «Стилистические особенности», такие как атрибуты или перечни пробелов с разделителями, могут также влиять на размер закодированного материала и сложность его обработки, поэтому использование таких особенностей это не только вопрос стиля. Если такие вопросы существенны, можно отдать предпочтение EXTENDED-XER перед BASIC-XER или CXER. В разделе 7 специфицируется BASIC-XER кодирование типов ACH.1. В разделе 8 специфицируется CXER кодирование типов ACH.1. В разделе 9 специфицируется EXTENDED-XER кодирование типов ACH.1 с использованием ссылок на более поздние разделы, которые определяют команды XER кодирования. В разделах с 10-го по 13-й перечисляются и классифицируются команды XER кодирования и специфицируют синтаксис для их назначения типам или компонентам ACH.1, с использованием либо префикса типа XER (см. ИСО/МЭК 8824-1, 30.3), либо управляющей секции XER кодирования. В разделе 14 определяется порядок предшествования, если команды XER кодирования присутствуют и в префиксе типа XER, и в управляющей секции XER кодирования. В разделе 15 специфицируется поддержка команд XER кодирования для пространства имен XML при использовании EXTENDED-XER. В разделе 16 специфицируется EXTENDED-XER кодирование. В разделах с 17 по 39 специфицируются: а) синтаксис каждой команды XER кодирования, используемой в префиксе типа или управляющей секции кодирования; б) ограничения команд XER кодирования, которые могут быть ассоциированы с конкретным типом ACH.1 (следствие наследования и множественных назначений); в) модификации правил XER кодирования, которые необходимы в EXTENDED-XER кодировании, когда применяются команды XER кодирования. Приложение А информативно и содержит примеры BASIC-XER и CXER кодирования. Приложение Б информативно и содержит описание частичного XML контента, который создан когда конструкции такие, как последовательность и последовательность-из утратили свои окружающие теги вместе с ограничениями спецификаций, облегчающими определение компонента ACH.1, с которым связан элемент XML. Приложение В информативно и содержит примеры команд XER кодирования и соответствующих результатов EXTENDED-XER кодирования.

Продолжение таблицы ГА.1

Раздел, подраздел, пункт, подпункт, таблица, приложение	Внесенные изменения
Раздел 1	Заменено «Настоящий стандарт устанавливает набор базовых правил XML кодирования (XER), которые могут быть использованы для получения синтаксиса перехода для значений типов, определенных в ИСО/МЭК 8824-1 и ИСО/МЭК 8824-2. Настоящий стандарт также определяет набор канонических правил XML кодирования, который предусматривает ограничения на базовые правила XML кодирования и производит уникальное кодирование для любого данного значения ACH.1» на «Настоящий стандарт устанавливает набор базовых правил XML кодирования (BASIC-XER), которые могут быть использованы с целью получения синтаксиса перехода для значений типов, определенных в ИСО/МЭК 8824-1 и ИСО/МЭК 8824-2. Настоящий стандарт определяет также набор канонических правил XML кодирования (CXER), которые предусматривают ограничения на базовые правила XML кодирования и производят уникальное кодирование для любого данного значения ACH.1. Настоящий стандарт специфицирует набор расширенных правил XML кодирования (EXTENDED-XER), добавляющий дополнительные опции кодера и позволяющий спецификатору ACH.1 варьировать кодирование, которое должно быть произведено в соответствии с BASIC-XER»
Раздел 1	Добавлено «Настоящий стандарт устанавливает также синтаксис и семантику команд XER кодирования и правила для их назначения и сочетания. Команды XER кодирования могут быть использованы для управления расширенными правилами XML кодирования (EXTENDED-XER) для конкретных типов ACH.1»
Раздел 2.1	Заменено «2.1 Идентичные рекомендации и международные стандарты В настоящем стандарте использованы нормативные ссылки на следующие рекомендации МСЭ-Т и международные стандарты: Рекомендация МСЭ-Т X.680 (2002) (ИСО/МЭК 8824-1:2002) Информационные технологии — абстрактная синтаксическая нотация версии 1 (ACH.1): Спецификация основной нотации Рекомендация МСЭ-Т X.681 (2002) (ИСО/МЭК 8824-2:2002) Информационные технологии — абстрактная синтаксическая нотация версии 1 (ACH.1): Спецификация информационного объекта Рекомендация МСЭ-Т X.682 (2002) (ИСО/МЭК 8824-3:2002) Информационные технологии — абстрактная синтаксическая нотация версии 1 (ACH.1): Спецификация ограничений Рекомендация МСЭ-Т X.683 (2002) (ИСО/МЭК 8824-4:2002) Информационные технологии — абстрактная синтаксическая нотация версии 1 (ACH.1): Параметризация ACH.1 спецификаций Рекомендация МСЭ-Т X.690 (2002) (ИСО/МЭК 8825-1:2002) Информационные технологии — Правила кодирования ACH.1: Спецификация базовых правил кодирования (BER), канонических правил кодирования (CER) и отличительных Правила Кодирования (DER) Рекомендация МСЭ-Т X.693 (2002) (ИСО/МЭК 8825-2:2002) Информационные технологии — Правила кодирования ACH.1: Спецификация правил уплотненного кодирования (PER)» на «2.1 Идентичные стандарты ИСО/МЭК 8824-1:2002, Информационные технологии – Абстрактная Синтаксическая Нотация Один (ACH.1): Спецификация базовой нотации. ИСО/МЭК 8824-1:2002/Amd.1:2003, Информационные технологии — Абстрактная Синтаксическая Нотация Один (ACH.1): Спецификация базовой нотации — Изменение 1: Поддержка Расширенных правил XML кодирования (EXTENDED-XER). ИСО/МЭК 8824-2:2002, Информационные технологии — Абстрактная Синтаксическая Нотация Один (ACH.1): Спецификация информационного объекта.

Продолжение таблицы ГА.1

Раздел, подраздел, пункт, подпункт, таблица, приложение	Внесенные изменения
	ИСО/МЭК 8824-2:2002/Amd.1:2003, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1) — Изменение 1: Спецификация информационного объекта — Поддержка Расширенных правил XML кодирования (EXTENDED-XER). ИСО/МЭК 8824-3:2002, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1): Спецификация ограничений. ИСО/МЭК 8824-4:2002, Информационные технологии — Абстрактная Синтаксическая Нотация Один (АСН.1): Параметризация спецификаций АСН.1. ИСО/МЭК 8825-1:2002, Информационные технологии — Правила кодирования АСН.1: Спецификация Базовых правил кодирования (BER), Канонических правил кодирования (CER) и Отличительных правил кодирования (DER). ИСО/МЭК 8825-1:2002/Amd.1:2003, Информационные технологии — Правила кодирования АСН.1: Спецификация Базовых правил кодирования (BER), Канонических правил кодирования (CER) и Отличительных правил кодирования (DER) — Изменение 1: Поддержка Расширенных правил XML кодирования (EXTENDED-XER). ИСО/МЭК 8825-2:2002, Информационные технологии — Правила кодирования АСН.1: Спецификация Правил уплотненного кодирования (PER). ИСО/МЭК 8825-2:2002/Amd.1:2003, Информационные технологии — Правила кодирования АСН.1: Спецификация Правил уплотненного кодирования (PER) — Изменение 1: Поддержка Расширенных правил XML кодирования (EXTENDED-XER). ИСО/МЭК 8825-3:2002, Информационные технологии — Правила кодирования АСН.1: Спецификация Управляющей нотации кодирования (ECN)»
Раздел 2.2	заменено «ИСО/МЭК 10646-1:1993 Информационные технологии — Универсальный многооктетный кодовый набор символов (UCS) — Часть 1: Архитектура и основная многоязычная матрица ИСО/МЭК 10646-1:1993/Изм.2:1996 Информационные технологии — Универсальный многооктетный кодовый набор символов (UCS) — Часть 1: Архитектура и основная многоязычная матрица — Изменение 2: UCS Формат Преобразования 8 (UTF-8) W3C XML 1.0:2000 Расширяемый язык разметки (XML) 1.0 (второе издание), рекомендации W3C, Copyright © [6 October 2000] World Wide Web Consortium, (Massachusetts Institute of Technology,Institut National de Recherche en Informatique et en Automatique, Keio University), gost.e-ecolog.ru REC-xml-20001006» на «– IETF RFC 2045 (1996), Многоцелевые расширения Интернет почты (MIME) Часть 1: Формат Совокупности Интернет сообщений. – IETF RFC 2141 (1997), Синтаксис Унифицированного имени ресурса (URN). – IETF RFC 2396 (1998), Унифицированные идентификаторы ресурса (URI): Типовой синтаксис. – IETF RFC 3061 (2001), Пространство имен URN объектных идентификаторов – ИСО/МЭК 10646-1:2000, Информационные технологии — Универсальный многооктетный набор кодированных символов (UCS) – Часть 1: Архитектура и базовая многоязычная матрица. – The Unicode Standard, Version 3.2.0, Консорциум Юникода. Примечание — Графические символы (и их коды), определенные в вышеуказанном стандарте идентичны тем, которые определены стандартом ИСО/МЭК 10646-1, но вышеуказанный стандарт включен в перечень, поскольку он также специфицирует имена управляющих символов. – W3C XML 1.0:2000, Расширяемый Язык Разметки (XML) 1.0 (Второе издание), Рекомендация W3C, Copyright © [6 октября 2000 г.] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), gost.e-ecolog.ru

Продолжение таблицы ГА.1

Раздел, подраздел, пункт, подпункт, таблица, приложение	Внесенные изменения
	– W3C XML Namespaces:1999, Пространство имен в XML, Рекомендация W3C, Copyright © [14 января 1999 г.] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), gost.e-ecolog.ru Примечание — Ссылка на документ из настоящего стандарта не дает ему, как автономному документу, статус международного стандарта»
Раздел 3	Заменено «В настоящем стандарте применяют следующие термины с соответствующими определениями» на «Для настоящего стандарта применимы определения ИСО/МЭК 8824-1, а также следующие определения»
Подраздел 3.2.1	Добавлены подразделы 3.2.1.1, 3.2.1.2, 3.2.1.3, и 3.2.1.4
Подраздел 3.2.2	Добавлены подразделы с 3.2.2.1 по 3.2.2.16
Подраздел 3.2.3	Заменено «правильный XML документ (для АСН.1 схемы) (valid XML document (for an ASN.1 schema)): XML документ, который правильно сформирован (см. W3C XML 1.0) и содержание которого соответствует XER спецификации правил кодирования XML для типов АСН.1, заданный АСН.1 схемой» на «действительный XML документ (для АСН.1 схемы) (valid XML document (for an АСН.1 schema)): XML документ, который правильно построен (см. W3C XML 1.0) и чье содержание соответствует спецификации BASIC-XER, CXER или EXTENDED-XER для кодирования типа АСН.1, заданного АСН.1 схемой, возможно включая команды XER кодирования»
Подраздел 3.2.3	Добавлены подразделы с 3.2.3.1 по 3.2.3.4
Раздел 3	Добавлены подразделы 3.2.5 и 3.2.6
Раздел 4	Добавлено «CXER — канонические правила XML кодирования; URI — универсальный идентификатор ресурса»
Раздел 5	Удален
Разделы 6—9	Перенумерованы в разделы 5—8
Подраздел 5.1	Заменено «CANONICAL-XER» на «CXER»
Подраздел 5.1	Добавлено «- расширенные правила XML кодирования (EXTENDED-XER)»
Подраздел 5.1	Удалено примечание
Подраздел 5.2	Заменено «Первый набор правил кодирования BASIC-XER в общем случае не производит канонического кодирования» на «Базовый набор правил кодирования BASIC-XER в общем случае не производит канонического кодирования и не обеспечивает любого пользователя управлением стилем XML, который он продуцирует»
Подраздел 5.3	Заменено «CANONICAL-XER» на «CXER» по всему тексту подраздела
Подраздел 5.4	Заменено «Если тип данных, закодированный по CANONICAL-XER, содержит типы EMBEDDED PDV, EXTERNAL или CHARACTER STRING, то внешнее кодирование перестает быть каноническим если кодирование,

Продолжение таблицы ГА.1

Раздел, подраздел, пункт, подпункт, таблица, приложение	Внесенные изменения
	используемое для всех типов EMBEDDED PDV, EXTERNAL или CHARACTER STRING, не является каноническим» на «Третий набор правил кодирования, установленный в настоящем стандарте — расширенные правила кодирования (EXTENDED-XER). Он определяется как вариации базовых правил кодирования BASIC-XER, установленные командами XER кодирования (см. 5.6), связанными с типом ACH.1. В отсутствие команд XER кодирования кодирование по расширенным правилам (EXTENDED-XER) отличается от кодирования по базовым правилам BASIC-XER только тем, что оно обеспечивает больше опций кодера»
Раздел 5	Добавлены подразделы 5.5 и 5.6
Раздел 6.1	Заменено «Динамическое соответствие базовых правил XML кодирования определено в разделе 8, динамическое соответствие для канонических правил XML кодирования — в разделе 9» на «Динамическое соответствие для базовых правил XML кодирования специфицируется в разделе 7, динамическое соответствие для канонических правил XML кодирования специфицируется в разделе 8 и динамическое соответствие для расширенных правил XML кодирования специфицируется в разделе 9»
Раздел 6.3	Заменено «Альтернативное кодирование разрешено базовыми правилами XML кодирования как опция кодера. Декодеры, соответствующие правилам кодирования XER, должны поддерживать все альтернативы» на «Альтернативное кодирование разрешается базовыми правилами XML кодирования и расширенными правилами XML кодирования как опция кодера. Декодеры, соответствующие BASIC-XER, должны поддерживать все альтернативы кодирования BASIC-XER. Декодеры, соответствующие EXTENDED-XER, должны поддерживать все альтернативы кодирования EXTENDED-XER. Примечание — Этот раздел применяется независимо от наличия (или отсутствия) каких-либо окончательных команд кодирования»
Подраздел 7.1	Заменено «8.1 Продукционные правила полного BASIC-XER кодирования 8.1.1 Соответствующий результат BASIC-XER кодирования — правильный XML документ, состоящий из: а) XML пролога (который может быть пустым) как определено в 8.2; б) элемента XML документа, который является законченным результатом кодирования значения единственного типа ACH.1, как определено в 8.3. 8.1.2 Спецификация в подразделах с 8.2 по 8.6 полностью определяет BASIC-XER кодирование. Примечание — Другие структурные компоненты, такие как команды XML обработки не разрешены этими подразделами и они никогда не создаются кодерами, соответствующими BASIC-XER. 8.1.3 XML документ должен быть кодирован с использованием UTF-8 для получения «строки октетов», которая формирует кодирование, в соответствии с настоящим стандартом. ACH.1 идентификатор объекта для этих правил кодирования определен в разделе 40. 8.1.4 В случае, если настоящий стандарт использует термин «разделитель» ("white-space") подразумевается один или несколько следующих символов стандарта Unicode: HORIZONTAL TABULATION (9), LINE FEED (10), CARRIAGE RETURN (13), SPACE (32). Цифры в круглых скобках — десятичные значения символов стандарта Unicode. Номер и выбор символа, который составляет разделитель — опция кодера» на «7.1 Продукционные правила полного BASIC-XER кодирования 7.1.1 Соответствующий результат BASIC-XER кодирования — правильный XML документ, состоящий из: а) XML пролога (который может быть пустым) как определено в 7.2;

Продолжение таблицы ГА.1

Раздел, подраздел, пункт, подпункт, таблица, приложение	Внесенные изменения
	б) элемента XML документа, который является законченным результатом кодирования значения единственного типа АСН.1, как определено в 7.3. 7.1.2 Спецификация в подразделах с 7.2 по 7.6 полностью определяет BASIC-XER кодирование. Примечание — Другие структурные компоненты, такие как команды XML обработки не разрешены этими подразделами и они никогда не создаются кодерами, соответствующими BASIC-XER. 7.1.3 XML документ должен быть кодирован с использованием UTF-8 для получения «строки октетов», которая формирует кодирование в соответствии с настоящим стандартом. Идентификатор объекта АСН.1 для этих правил кодирования определен в разделе 39. 7.1.4 Если в настоящем стандарте используется термин «пробел» ("white-space"), подразумевается один или несколько следующих символов стандарта Unicode: HORIZONTAL TABULATION (9), LINE FEED (10), CARRIAGE RETURN (13), SPACE (32). Цифры в круглых скобках — десятичные значения символов стандарта Unicode. Количество и выбор символов, которые составляют пробел — опция кодера»
Подраздел 7.1	Добавлен Подраздел 7.1.5
Подраздел 7.3.1	Заменено «Элемент XML документа должен быть "XMLTypedValue", как определено в ИСО/МЭК 8824-1, пункт 15.2, с изменениями и ограничениями, указанными в следующих подпунктах» на «Элементом XML документа будет "XMLTypedValue", как определено в ИСО/МЭК 8824-1, пункт 15.2, с изменениями и ограничениями, установленными в последующих подразделах раздела 7.3»
Подраздел 7.3.1	Добавлен подраздел 7.3.1.1
Подраздел 7.3.3	Заменено «В случаях, если ИСО/МЭК 8824-1 допускает использование разделителя АСН.1 между лексическими элементами, символы должны быть ограничены разделителями, определенными в 8.1.4» на «Когда ИСО/МЭК 8824-1, пункты 11.1.4, 11.11 и 11.13 разрешают использование пробела АСН.1 между лексическими единицами в "xmlbstring" или в "xmlhstring", использованные символы должны быть ограничены до "разделителей", определенных в 8.1.4» на «Если ИСО/МЭК 8824-1, 11.1.4, 11.11 и 11.13 разрешают пробелы АСН.1 между лексическими единицами в "xmlbstring" или в "xmlhstring", использованные символы должны быть ограничены до «разделителей», определенных в 7.1.4»
Подраздел 7.3.3	Добавлен подраздел 7.3.3.1
Подраздел 7.3.4	Заменено «Значение "XMLIntegerValue", определенное в ИСО/МЭК 8824-1, пункт 18.9, должно быть только "SignedNumber" (числом со знаком)» на "XMLIntegerValue" определенный в ИСО/МЭК 8824-1, 18.9, должен быть только "XMLSignedNumber"
Подраздел 7.3.4	Добавлены подразделы 7.3.4.1 и 7.3.4.2
Раздел 7	Добавлены подразделы 7.5 и 7.6
Подраздел 8.1	Заменено «Общие правила для канонического XML кодирования» на «Общие правила для канонического XER»

Продолжение таблицы ГА.1

Раздел, подраздел, пункт, подпункт, таблица, приложение	Внесенные изменения
Подраздел 8.1.2	Заменено «Все лексические элементы, формирующие "XMLTypedValue", не должны иметь между собой разделитель (см. ИСО/МЭК 8824-1, подпункт 11.1.4)» на «Все лексические единицы, формирующие "XMLTypedValue", не должны иметь пробел ("white-space") между ними (см.7.3.3)»
Подраздел 8.3.1	Заменено «Если "XMLTypedValue" можно использовать вместо "XMLBitStringValue" (см. ИСО/МЭК 8824-1, пункт 21.9), как определено в ИСО/МЭК 8824-1, пункт 21.10, то его следует использовать. В противном случае, "xmlbstring" может использоваться только без разделителей» на «Если альтернатива "XMLTypedValue" для "XMLBitStringValue" (см. ИСО/МЭК 8824-1, 21.9) может быть использована (как определено в ИСО/МЭК 8824-1, 21.10), то она должна быть использована. В противном случае должна быть использована альтернатива "xmlbstring" со всеми удаленными пробелами "white-space" (см. 7.3.3)»
Подраздел 8.4	Заменено «Если "XMLTypedValue" можно использовать вместо "XMLOctetStringValue" (см. ИСО/МЭК 8824-1, пункт 22.3) как определено в ИСО/МЭК 8824-1, пункт 22.4, то его следует использовать. В противном случае, "xmlhstring" может использоваться только без разделителей, и все символы должны быть в верхнем регистре» на «Если альтернатива "XMLTypedValue" для "XMLOctetStringValue" (см. ИСО/МЭК 8824-1, 22.3) может быть использована (как определено в ИСО/МЭК 8824-1, 22.4), то она должна быть использована. В противном случае должна быть использована альтернатива "xmlbstring" со всеми удаленными пробелами "white-space" (см. 7.3.3) и все буквы должны быть заглавными»
Подраздел 8.6.1	Заменено «Тип «множество» должен содержать элементы в его "RootComponentTypeList", отсортированные в каноническом порядке (как определено в ИСО/МЭК 8824-1, пункт 8.6). В целях определения порядка кодирования, в случае если имеется компонент с типом выбора «без применения тегов», он должен быть упорядочен, также как и в случае с тегом, равным наименьшему тегу в "RootAlternativeTypeList" этого типа выбора или любого вложенного типа выбора «без применения тегов»» на «Тип набора будет иметь элементы в его "RootComponentTypeList", отсортированном в каноническом порядке, определенном в ИСО/МЭК 8824-1, 8.6, и дополнительно с тегом ACH.1 для целей определения порядка, в котором компоненты закодированы, когда один или более компонентов является типом выбора. Каждый такой тип выбора расположен по порядку так, как если бы он имел тег, равный наименьшему тегу в "RootAlternativeTypeList" того типа выбора или любого такого типа выбора, который встроен в него»
Подраздел 8.7.1	Заменено «Порядок элементов "XMLSetOfValue" (см. ИСО/МЭК 8824-1, пункт 27.3) должен быть определен сортировкой символьных строк, которые представляют собой CANONICAL-XER кодирование для каждого элемента, в соответствии с 9.7.2, 9.7.3» на «Порядок элементов "XMLSetOfValue" (см. ИСО/МЭК 8824-1, 27.3) будет определяться сортировкой строк символов, которую обеспечивает CXER кодирование для каждого элемента, как определено в 8.7.2 и 8.7.3»
Раздел 8	Добавлен подраздел 8.12
Структура стандарта	Добавлены разделы 9—38
Раздел 10	Перенумерован в раздел 39

Окончание таблицы ГА.1

Раздел, подраздел, пункт, подпункт, таблица, приложение	Внесенные изменения
Подраздел 39.2	Заменено «Для CANONICAL-XER» на «Для CXER»
Подраздел 39.2	Добавлено «Для EXTENDED-XER: {joint-iso-itu-t ACH1 (1) xml-encoding (5) extended (2)} "Extended XML encoding of a single ACH.1 type"»
Раздел 39	Добавлен подраздел 39.3
Приложение А	Заменено «Примеры кодирования» на «Примеры кодирования BASIC-XER и CXER»
Приложение А	Заменено «Настоящее приложение иллюстрирует применение правил XML кодирования, а также содержит XML разметку представления (гипотетической) записи данных о персонале, которая определяется с использованием ACH.1.» на «Данное приложение иллюстрирует применение базовых и канонических правил XML кодирования, определенных в настоящем стандарте. Приложение содержит XML разметку представления (гипотетической) записи данных о персонале, которая определена с использованием ACH.1.»
Приложение А.1	Заменено «Примечание — Теги используются только для сравнения с примером предыдущей версии ИСО/МЭК 8824-1 и не влияют на XML кодирование» на «Примечание — В этом примере используются теги только для сравнения с примером ранней версии ИСО/МЭК 8824-1. Они не влияют на BASIC-XER, CXER и EXTENDED-XER XML кодирование»
Структура стандарта	Добавлены приложения Б и В

Приложение ГБ
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ГБ.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 8824-1:2002	—	*
ИСО/МЭК 8824-1:2002/Amd.1:2003	—	*
ИСО/МЭК 8824-2:2002	—	*
ИСО/МЭК 8824-2:2002/Amd.1:2003	—	*
ИСО/МЭК 8824-3:2002	—	*
ИСО/МЭК 8824-4:2002	—	*
ИСО/МЭК 8825-1:2002	—	*
ИСО/МЭК 8825-2:2002		*
ИСО/МЭК 8825-2:2002/Amd.1:2003	—	*
ИСО/МЭК 8825-3:2002	—	*
ИСО/МЭК 10646-1:1993	—	*
ИСО/МЭК 10646-1:1993/Amd.2:1996	—	*

* Соответствующий межгосударственный стандарт отсутствует. До его принятия рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.

УДК 681.324:006.354 ОКС 35.100.60 П80 ОКП 50 1300

Ключевые слова: обработка данных, информационный обмен, сетевое взаимодействие, взаимосвязь открытых систем, коммуникационная процедура, преобразование данных, кодирование, правила (инструкции), XML

Редактор *Н. Б. Лурье*
Технический редактор *Н. С. Гришанова*
Корректор *С. В. Смирнова*
Компьютерная верстка *Т. Ф. Кузнецовой*

Сдано в набор 21.01.2011. Подписано в печать 27.04.2011. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 10,70. Уч.-изд. л. 9,72. Тираж 109 экз. Зак. 49

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.

gost.e-ecolog.ru