



中华人民共和国国家标准

GB/T 16262.3—2006/ISO/IEC 8824-3:2002

信息技术 抽象语法记法—(ASN. 1) 第 3 部分:约束规范

Information technology—Abstract Syntax Notation One(ASN. 1)—
Part 3:Constraint specification

(ISO/IEC 8824-3:2002, IDT)

2006-03-14 发布

2006-07-01 实施

中华人民共和国国家质量监督检验检疫总局 发布
中国国家标准化管理委员会

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
3.1 基本记法规范	1
3.2 信息客体规范	1
3.3 ASN.1 规范的参数化	1
3.4 附加定义	1
4 缩略语	2
5 约定	2
6 记法	2
6.1 赋值	2
7 ASN.1 词项	2
7.1 附加关键字	2
8 一般约束规范	2
9 用户定义的约束	3
10 表约束,包括成分关系约束	4
11 内容约束	7
附录 A(规范性附录) 约束的单一实例类型	8
附录 B(资料性附录) 记法综述	9

前 言

GB/T 16262 在《信息技术 抽象语法记法一(ASN.1)》总标题下,目前包括以下 4 个部分:

第 1 部分(即 GB/T 16262.1):基本记法规范;

第 2 部分(即 GB/T 16262.2):信息客体规范;

第 3 部分(即 GB/T 16262.3):约束规范;

第 4 部分(即 GB/T 16262.4):ASN.1 规范的参数化。

本部分为 GB/T 16262 的第 3 部分,等同采用国际标准 ISO/IEC 8824-3:2002《信息技术 抽象语法记法一(ASN.1):约束规范》(英文版)。与该项国际标准的等同文本是 ITU-T 建议 X.682。

按照 GB/T 1.1—2000 的规定,本部分对 ISO/IEC 8824-3:2002 作了下列编辑性修改:

——“本标准”一词改为“本部分”;

——在引用的标准中,凡已转化成我国标准的各项标准,均用我国的相应标准编号代替。

本部分的附录 A 是规范性附录,附录 B 是资料性附录。

本部分由中华人民共和国信息产业部提出。

本部分由中国电子技术标准化研究所归口。

本部分起草单位:中国电子技术标准化研究所。

本部分主要起草人:郑洪仁、安金海、徐云驰。

引 言

应用设计者需要一种定义结构式数据类型的记法以传送其语义。这在 GB/T 16262.1 和 GB/T 16262.2 中提供。还需要一种记法以进一步约束可能出现的值。这些约束的示例有限制某个(某些)成分的范围或使用规定的信息客体集合约束“ObjectClassFieldType”成分,或使用“AtNation”规定各成分之间的关系。

GB/T 16262 的本部分对约束规范的一般情况提供记法。

注 1: 由于历史原因,“子类型约束”的特殊情况在 GB/T 16262.1 中规定。

约束记法可能出现在使用语义结构“Type”之后的圆括号中,本部分的目的是要规定圆括号中的一般情况。

注 2: 对相同的“Type”可以应用多重约束(每个在其自身的圆括号之内),如同约束“Type”的结果在形式上本身就是“Type”结构。

将一个约束应用于“Type”结构的最外层时,就会产生一个新的类型,它是原(双亲)类型的子类型。

在约束记法的其他使用场合,双亲类型的子类型本身可用来定义相同双亲类型的其他子类型。因此,可通过限制双亲类型的范围或将此子类型规定成值的并集来定义构成子类型的值的子集。

注 3: 在 GB/T 16262.1—2006 的 15.7 中规定的“ValueSet”记法提供了规定子类型的更进一步的办法。

当此记法嵌入另一种类型时,约束还可用来产生双亲类型的子类型(如上所述)。虽然,某些“成分关系”约束放在“Type”之后(在集合或序列类型定义中),但是,不能用来限制它们之前的“Type”(引用成分)可能值的集合。而是它们规定引用成分的值与同一集合或序列类型中一个或几个其他“Type”(称为引用成分)的值之间的关系。

成分关系约束可以看成将它们所嵌入的序列类型分成子类,而不必看成引用类型。

通过使用信息客体集合限制成分中的类型或值时可以应用有关 ObjectClassFieldType 成分的约束。这种约束称为表约束,因为它们是根据客体集合的“关联表”规定的。本部分定义的成分关系约束是表约束的特殊情况。

最后,可通过规定子类型中的人可读文本方式值集合,将“Type”划分成子类型。这种约束称为用户定义的约束。例如,规定用户定义的约束可以是比特串约束到规定的 ASN.1 类型的值加密所产生的值集合。

本部分的目的是要提供规定表约束(包括成分关系约束)和用户定义的约束所用的记法。

注 4: 一般而言,用灵活的方式,对约束规范的全面支持(特别是成分关系约束、划分子类型约束和形式定义体中用户定义约束)会要求一种能力可与程序设计语言相当的记法。显然这种能力只能通过建立 ASN.1 记法与某个其他定义的计算机语言之间的链接来提供。本部分这一版不提供这种链接,因此只能支持少量的约束机制。

嵌入定义约束的记法(子类型和关系)往往是最方便的规范形式(特别对结构的原成分的简单子分类),但是,有时将优先采用单独的(外部的)规范,特别是,由定义基本协议的一个独立小组要强加的约束就是如此。

注 5: 在 GB/T 16262.4 中定义的参数化专门设计成能使一部分 ASN.1 规范(以及特别是约束)被参数化,允许某个为参数化构造提供实际参数的其他小组施加实际约束。

本部分所支持的约束规范所用的记法是：

- 用户定义的约束(见第 9 章)；
- 表约束,包括运载与信息客体有关的值、使用 GB/T 16262.2 的记法定义的两个成分之间的成分关系约束(见第 10 章)；
- 内容约束(见第 11 章)。

表约束对 GB/T 16262.2—2006 附录 C 的“InstanceOfType”构造的应用在本部分附录 A 中规定。

信息技术 抽象语法记法一(ASN.1)

第3部分:约束规范

1 范围

GB/T 16262的本部分是抽象语法记法一(ASN.1)的一个部分,并提供规定用户定义的约束、表约束和内容约束的记法。

2 规范性引用文件

下列文件中的条款通过 GB/T 16262 的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 16262.1—2006 信息技术 抽象语法记法一(ASN.1) 第1部分:基本记法规范(ISO/IEC 8824-1:2002,IDT)

GB/T 16262.2—2006 信息技术 抽象语法记法一(ASN.1) 第2部分:信息客体规范(ISO/IEC 8824-2:2002,IDT)

GB/T 16262.4—2006 信息技术 抽象语法记法一(ASN.1) 第4部分:ASN.1规范的参数化(ISO/IEC 8824-4:2002,IDT)

3 术语和定义

下列术语和定义适用于 GB/T 16262 的本部分。

3.1 基本记法规范

本部分使用 GB/T 16262.1—2006 中定义的术语。

3.2 信息客体规范

本部分使用 GB/T 16262.2—2006 中定义的术语。

3.3 ASN.1 规范的参数化

本部分使用 GB/T 16262.4—2006 中定义的下列术语:

——参数化类型 parameterized type。

3.4 附加定义

3.4.1

成分关系约束 component relation constraint

关于集合类型或序列类型值的约束,其在文本上与此集合类型或序列类型的成分的类型(引用的成分)之一相关,并且规定那个成分的值与一个或多个其他成分(被引用成分)的值之间的关系。

3.4.2

被约束类型 constrained type

包含引用的成分和某个成分关系约束中所有被引用成分的最内的“Type”。

3.4.3

约束集合 constraining set

在某个成分关系约束中被引用的信息客体集合。

3.4.4

约束表 constraining table

与约束集合相对应的关联表(见 GB/T 16262.2—2006 的第 13 章)。

3.4.5

被引用成分 referenced component

在成分关系约束中被标识的集合类型或序列类型的成分。

3.4.6

引用的成分 referencing component

具有相关的成分关系约束的集合类型或序列类型的成分。

3.4.7

被选行 selected rows

在适当的列中,约束表包含所有被引用成分的值的某些行。

3.4.8

表约束 table constraint

应用于要求其值符合某个表合适列的内容的客体类别字段类型的约束。

3.4.9

用户定义的约束 user-defined constraint

要求比能由其他约束形式涉及的语句更为复杂的语句的约束,因此必须包含 ASN.1 之外的其他规范。

4 缩略语

本部分使用下列缩略语:

ASN.1 抽象语法记法一

5 约定

本部分采用 GB/T 16262.1—2006 的第 5 章定义的记法约定。

6 记法

本章综述本部分定义的记法。

6.1 赋值

本部分定义了下列记法,这个记法能用作“ConstraintSpec”(见 GB/T 16262.1—2006 的 45.6)的替代记法:

——GeneralConstraint (见 8.1)。

7 ASN.1 词项

除 GB/T 16262.1—2006 第 11 章中规定的词汇之外,本部分中还使用下列各条规定的词项。适用于这些词项的一般规则是按 GB/T 16262.1—2006 的 11.1 定义的。这些新的词项使用 GB/T 16262.1—2006 第 10 章规定的 ASN.1 字符集。

7.1 附加关键字

名称 CONSTRAINED、CONTAINING、ENCODED 和 BY 已在 GB/T 16262.1—2006 的 11.27 中列出。

8 一般约束规范

8.1 “GeneralConstraint”的记法如下:

```

GeneralConstraint ::=
    UserDefinedConstraint |
    TableConstraint |
    ContentsConstraint

```

8.2 约束规范的各种可能性定义如下:

- a) “UserDefinedConstraint”在第 9 章中定义;
- b) “TableConstraint”在第 10 章中定义;
- c) “ContentsConstraint”在第 11 章中定义。

9 用户定义的约束

注 1: 这种形式的约束规范可以看作是 ASN.1 说明的特殊形式,因为它不是可由机器完全处理的。然而,使用存在的特定用户定义的约束,调用由用户提供的约束检验程序的自动工具却是可能的。

注 2: 协议的设计者要知道由于这种方法的约束定义不是可由机器完全处理的,因此采用这种能力的规范由自动工具处理可能不太容易。

9.1 用户定义的约束由下列语法规定:

```

UserDefinedConstraint ::=
    CONSTRAINED BY " { " UserDefinedConstraintParameter ", " * " } "

```

9.2 建议在括号 (“{”和“}”)内注释何处引用实际约束。此注释应清楚地表明“UserDefinedConstraint”施加什么约束。

注: 如果在此括号中存在“UserDefinedConstraintParameter”(见 9.3),此注释可以在它们之前、之后或分散在它们之中,这要随设计者方便而定。

9.3 应用的实际约束可能取决于某些参数。对每一个这种参数,应在“UserDefinedConstraint”中包含“UserDefinedConstraintparameter”。每个“UserDefinedConstraintParameter”应是成行定义或是引用名的“Value”、“ValueSet”、“Object”、“ObjectSet”、“Type”或“DefinedObjectClass”。

注: 如果在“ParameterizedAssignment”使用“UserDefinedConstraint”,则引用名可以是假设的参数。

```

UserDefinedConstraintParameter ::=
    Governor " ; " Value |
    Governor " ; " ValueSet |
    Governor " ; " Object |
    Governor " ; " Objec Set |
    Type |
    DefinedObjectClass

```

记法“Governor”在 GB/T 16262.4—2006 的 8.3 中定义。当使用第一或第二种替代记法时,Governor”应是“Type”。当使用第三或第四种替代记法时,“Governor”应是“DefinedObjectClass”。

9.4 示例

如果一名应用设计者想将某些成分规定为比特串来承载某个 ASN.1 类型的值的密码(对每个成分名不相同),那么,(使用 GB/T 16262.4 的参数化时),参数化的 ENCRYPDED 类型可以定义如下:

```

ENCRYPDED{ToBeEnciphered} ::= BIT STRING
(CONSTRAINED BY
    {--必须是某个编码 BER 的密码之值
    ——ToBeEnciphered}
    ! Error ;securityViolation)
Error ::= ENUMERATED{SecurityViolation}

```


并且使用 BIT STRING(是 ENCRYPTED 类型的)的 ENCRYPTED 参数化子类型变得简单:

```
ENCRYPTED{SecurityParameters}
```

或在设计者灵机一动时,相当于:

```
BIT STRING(ENCRYPTED{SecurityParameters})
```

按照本地安全政策处理 SecurityViolation 的出现。

10 表约束,包括成分关系约束

注1: 信息客体类别、信息客体,信息客体集合和客体类别字段类型在 GB/T 16262.2 中定义。在本章中假定已了解了这些概念。

注2: 本章使用定义双亲类型的主要记法中所标识的信息客体集合来描述表约束的应用,换言之,主要记法是由协议设计者定义和标识的。这不能满足对实际信息客体集合的要求,因实际信息客体集合要用作随语法而变化的特定抽象语法的约束。GB/T 16262.4 另外提供记法,该记法使约束中使用的信息客体集合能作为一种参数,而变化的抽象语法组在后来的某一时间会支持这种参数值。

示例:

为说明本章的条文,将使用下面的示例。ErrorReturn 类型承载 errorCategory 和与此种类 error-Info 相对应的一个或几个 errorCode。这由 ERROR-CLASS 信息客体类别所支持。此客体类别具有信息客体集合 ErrorSet 所定义的特定客体集合,而 ErrorSet 是用来约束 ErrorReturn 的字段的。

我们有:

```
ERROR-CLASS;:=CLASS
{
    &-category PrintableString(SIZE(1)),
    &-code INTEGER,
    &-Type
}
WITH SYNTAX{&-category &-code &-Type}
ErrorSet ERROR-CLASS;:=
{
    {" A" 1 INTEGER} |
    {" A" 2 REAL} |
    {" B" 1 CHARACTER STRING} |
    {" B" 2 GeneralString}
}
ErrorReturn;:=SEQUENCE
{
    errorCategory ERROR-CLASS. &-category({ErrorSet})OPTIONAL,
    errors SEQUENCE OF SEQUENCE
{
    errorCode ERROR-CLASS. &-code({ ErrorSet}{@errorCategory}),
    errorInfo ERROR-CLASS. &-Type ({ErrorSet} {@errorCategory , @. errorCode})
}OPTIONAL
}
```

可将 Error Set 的关联表绘制如下:

&.category	&.code	&.Type
" A"	1	INTEGER
" A"	2	REAL
" B"	1	CHARACTER STRING
" B"	2	GeneralString

10.1 表约束只能应用于类型“ObjectClassFieldType”或“InstanceOfType”。前者在本章的其余各条中定义,后者在附录 A 中定义。

10.2 “ObjectClassFieldType”标识信息客体类别,并且是标识该类别允许的“FieldName”之一。表约束标识信息客体集合,其关联表(如 GB/T 16262.2—2006 的第 13 章中定义)确定被约束值的集合。

10.3 “TableConstraint”记法是:

```
TableConstraint ::=
    SimpleTableConstraint |
    ComponentRelationConstraint
SimpleTableConstraint ::= Object Set
```

10.4 “SimpleTableConstraint”中的“ObjectSet”由正被约束的“ObjectClassFieldType”中出现的类别支配。

10.5 “SimpleTableConstraint”的语义使用约束信息客体集合的关联表加以规定。

10.6 被约束类型的“FieldName”用来选择此关联表适用的列,然而,应用时尚需遵守下列规则:

- 对类型字段,此成分约束为此列任一行中任一类型的任一值;
- 对值字段,此成分约束为此列任一行中的任一值;
- 对值集合字段,此成分约束为此列任一行值集合中的任一值。

注:对某种给定客体集合,如果上述算法不能求得允许的值,如果此成分又存在于约束类型的值之中,那么这种约束总是违规的。

示例:

在第 10 章的示例中,情况 b) 适用于成分 ErrorCategory:

```
ErrorCategory ERROR-CLASS. &.category({ErrorSet}) OPTIONAL,
```

同时,ErrorSet 的关联表将其可能的值限制于“A”和“B”。

10.7 使用信息客体集合的关联表和下列生成式应用于成分关系约束:

```
ComponentRelationConstraint ::=
    " { " DefinedObjectSet " } " " { " At Notation " , " + " } "
AtNotation ::=
    " @ " ComponentIdList |
    " @ . " Level Component Id List
Level ::= " . " Level | empty
ComponentIdList ::= identifier " . " +
```

10.8 “ComponentIdList”中的每个“identifier”标识其双亲是集合、序列或选择类型的成分,如果它标识的成分不是集合、序列或选择类型,应是最后的“identifier”。

10.9 在双亲是集合或序列类型的情况,“identifier”应是此双亲“ComponentTypeLists”中“NamedType”的标识符之一。在双亲是选择类型的情况,“identifier”应是此选择类型“AlternativeTypeLists”中替代记法“NamedType”的“identifier”之一。

10.10 “AtNotation”向其所在的 ASN.1 结构的其他成分提供指针。“ComponentIdList”中第一个“identifier”的双亲结构被确定如下:

- a) 如果选择“*AtNotation*”的第一个替代记法(在“@”之后无“.”),那么双亲结构是最外的文本上封闭的集合类型、序列类型或选择类型。
- b) 如果选择第二个替代记法(在“@”之后有“.”),那么通过从最内的文本上封闭的集合类型或序列类型向上移动等于@之后附加“.”的数目的若干层次(集合、单一集合、单一序列、序列、选择)获得双亲结构。在出现“*AtNotation*”时,附加的“.”的个数不应超过包含最内的集合或序列类型的结构(集合、单一集合、单一序列、选择)的数目。

注:尽管在包含序列、单一序列、集合、单一集合和选择类型组合的结构中可能出现不同嵌套层次,但是只有文本上处于集合类型或序列类型并引用文本上处于同一集合或序列类型的某个其他字段时才允许“*AtNotation*”。

示例:

在下列示例中“@…”说明上面的 b)情况

```
ErrorMessage ::= SEQUENCE{
    severity ERROR. &severity({Errors}),
    parameters SEQUENCE OF SEQUENCE{
        errorId ERROR. &id({Errors}),
        data SEQUENCE OF SEQUENCE{
            value ERROR. &Type({Errors}){@ severity,@…errorId}),
            text visibleString}}
```

10.11 使用这种记法的成分是引用的成分,由“*AtNotation*”标识的成分是被引用的成分。

10.12 “*ObjectSet*”(见 10.3)或“*DefinedObjectSet*”(见 10.7)是约束集合,由此求得的关系表(在 GB/T 16262.2—2006 的第 13 章中规定)是约束表。

10.13 成分关系约束只能应用于文本上属于封闭“*Type*”(被约束类型)的 ASN.1 类型,而此封闭 *Type* 在文本上包含所有被引用成分。被约束类型定义为满足上述条件的最内的“*Type*”。

示例:

在第 10 章的示例中,被约束类型是 *ErrorReturn*。

注:在某些方面,将此约束的应用看成是使用被引用成分的值以标识约束表中的行,然后使用合适的列表来约束引用的成分是不可能的。据此观点,被引用成分本身不能被看成是被约束的。

然而,下面所用的方法略有不同。它将此约束看成是在被约束类型(如上述解释,它不是引用的成分的类型)所有可能值上的操作和选择一些满足此约束的值。这种方法使得讨论有关不包含引用成分,或不包含一个或几个被引用成分(因为它们是任选的或在选择中)的值的被约束类型的值,以及被引用成分之一具有不与约束表中任一行相对应的值的被约束类型的值成为可能。

10.14 要求引用和所有被引成分是引用同一类别的“*ObjectClassFieldType*”。要求约束的集合是这些类别的信息客体集合。要求被引用成分是由与引用的成分相同的客体集合的约束的值字段或值集合字段。

示例:

在第 10 章的示例中,“*ObjectClassFieldType*”是所有的类别 *ERROR-CLASS*,也是约束的集合,它是 *ErrorSet*。

10.15 下面几条确定被约束类型中满足此约束的值集合。

10.16 如果引用的成分不在被约束类型的某一值中,那么这个值满足此约束。

示例:

在第 10 章的示例中,如果没有成分 *error*,那么认为 *error* 上的约束被满足。

10.17 如果任一被引用成分不在被约束类型的某一值中,而这个值不满足此约束,除非引用的成分不存在,在这种情况下此约束总是被满足的。

10.18 如果所有被引用成分存在并且引用的成分也存在,那么不满足此约束,除非在约束表中存在一个或几个这样的被选行,对每个被选行:

- a) 值字段的每个被引用成分具有的值是被选行相对应的列的值；
- b) 值集合字段的每个被引用成分具有的值是被选行相对应的列的值集合之一的值。

10.19 只有在引用成分满足通过将只包含被选行的表应用于引用成分而获得的简单表约束(如上面定义的),那就满足此约束。

示例:

在第10章的示例中,成分 `errorCategory`、`errorCode` 和 `errorInfo` 必须对应于 `ErrorSet` 关联表的某一行。

10.20 如果“`ObjectClassFieldType`”受一个或几个“`TableConstrain`”的约束,并且“`FieldName`”表示类型字段、可变类型值字段或可变类型值集合字段,那么在每种通信实例中,如果被引用的成分之一是标识符字段,那么被选行的数目应是1,否则至少应选择1。

示例:

在第10章的示例中,如果还有一个客体{“B”2 `PrintableString`} ,那么可能有一个以上被选行。

11 内容约束

11.1 内容约束由下列语法规定:

```
ContentsConstraint ::=
    CONTAINING Type |
    ENCODED BY Value |
    CONTAINING Type ENCODED BY Value
```

11.2 “`Value`”应是类型客体标识符的值。

11.3 “`ContentsConstraint`”仅适用于八位位组串类型和不用“`NamedBitList`”定义的位串类型。这些被约束类型不应有用于它们的进一步约束,不管是直接或通过“`typereference`”名数的使用。

11.4 “`Contents Constraint`”的第一个生成式规定八位位组串或位串的抽象值是“`Type`”抽象值的编码,此编码是由适用于八位位组串或位串的编码规则产生的。下列限定使用:

- a) 假若此约束应用于八位位组串,如果“`Type`”的抽象值的编码不是八位位组的倍数,那么它是规范差错。
- b) 如果八位位组串或位串有长度约束,“`Type`”抽象值被约束为其编码能被包含在被约束八位位组串或位串中的值。如果没有这种抽象值,那么它是规范差错。

11.5 “`ContentsConstraint`”的第二生成式规定由客体标识符值“`Value`”标识的规程应产生并解释位串或八位位组串的内容。假若位串或八位位组串已受到约束,如果这些规程没有产生满足此约束的编码,那么它是规范差错。

11.6 “`ContentsConstraint`”的第三个生成式规定八位位组串或位串的抽象值是“`Type`”抽象值的编码,此编码是由客体标识符值“`Value`”所标识的编码规则产生的。下列限定适用:

- a) 假若此约束应用于八位位组串,如果“`Type`”的抽象值的编码不是八位位组的倍数,那么它是规范差错。
- b) 如果八位位组串或位串有长度约束,“`Type`”抽象值被约束为其编码能被包含在被约束八位位组串或位串中的值。如果没有这种抽象值,它是规范差错。

附 录 A
(规范性附录)
约束的单一实例类型

A.1 本附录规定对 GB/T 16262.2—2006 的附录 C 中规定的“InstanceOfType”的约束的应用。

A.2 适用于这种类型的仅有的约束是如第 10 章规定的简单表约束。当用此方法约束时,“InstanceOfType”的等效序列类型是:

```
SEQUUNCE
{
    type-id(DefinedObjectClass) . &-id({<DefinedObjectSet>}),
    value[o]< DefinedObjectClass>. &-Type({<Defined Object Set>}{@. type-id})
}
```

其中<DefinedObjectClass>由用于“InstanceOfType”记法的特定“DefinedObjectClass”代替,DefinedObjectSet>由用于简单表约束的特定“DefinedObjectSet”代替。

A.3 当多个约束应用于单一实例类型时,每一个产生一个上述形式的约束,以致有适用于等效序列类型每一元素的多个约束。

A.4 示例:

基于 GB/T 16262.2—2006 的 C.10“示例”的示例如下。

类型:

```
INSTANCE OF MHS-ODY-LASS({PossibleBodyTypes})
```

具有如下的等效序列类型:

```
[UNIVERSAL 8]IMPLICIT SEQUENCE
{
    type-id MHS-ODY-LASS. &-id({PossibleBodyTypes}),
    value[o] MHS-ODY-LASS. &-Type({PossibleBodyTypes} {@. type-id})
}
```

这里,序列的 type-id 成分限于取 PossibleBodyTypes 之一的 &-id 字段的值,而 Value 成分限于同一信息客体的 &-Type 字段的任一值。

在这种情况下,PossibleBodyTypes 多半是此规范的一个参数(见 GB/T 16262.4—2006 的第 10 章和 A.8),在产生协议实现一致性声明(PICS)之前,使上述约束成为 GB/T 16262.4—2006 的 10.3 定义的可变约束,不太可能解决此参数。

附 录 B
(资料性附录)
记 法 综 述

在 7.1 中定义了下列词项：

CONSTRAINED
CONTAINING
ENCODED
BY

在 GB/T 16262.1—2006 中定义了下列词项并用于本部分：

modulereference
number
":=" "
"{" "
"}" "
"," "
". "

在 GB/T 16262.1—2006 中定义了下列生成式并用于本部分：

Type
Value
ValueSet

在 GB/T 16262.2—2006 中定义了下列生成式并用于本部分：

DefinedObjectClass
DefinedObjectSet
Object
ObjectSet

在 GB/T 16262.4—2006 中定义了下列生成式并用于本部分：

Governor

本部分中定义了下列生成式：

GeneralConstraint ::= UserDefinedConstraint | TableConstraint | ContentsConstraint
UserDefinedConstraint ::= CONSTRAINED BY "{" UserDefinedConstraintParameter "," * "}"
UserDefinedConstraintParameter ::= =
 Governor ":" Value |
 Governor ":" ValueSet |
 Governor ":" Object |
 Governor ":" ObjectSet |
 Type |
 DefinedObjectClass
TableConstraint ::= SimpleTableConstraint | ComponentRelationConstraint
SimpleTableConstraint ::= ObjectSet
ComponentRelationConstraint ::= "{" DefinedObjectSet "}" "{" AtNotation "," + "}"
AtNotation ::= "@." ComponentIdList | "@." Level ComponentIdList

Level ::= "." Level | empty

ComponentIdList ::= identifier "." +

ContentsConstraint ::=

CONTAINING Type |

ENCODED BY Value |

CONTAINING Type ENCODED BY Value
