

中华人民共和国国家标准

GB/T 16264.6—2008/ISO/IEC 9594-6:2005
代替 GB/T 16264.6—1996

信息技术 开放系统互连 目录 第6部分：选定的属性类型

Information technology—Open Systems Interconnection—The Directory—
Part 6: Selected attribute types

(ISO/IEC 9594-6:2005 Information technology—Open Systems
Interconnection—The Directory: Selected attribute types, IDT)

2008-08-06 发布

2009-01-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

国家图书馆专用

目 次

前言	III
引言	IV
第一篇 概述	1
1 范围	1
2 规范性引用文件	1
3 术语和定义	2
4 约定	2
第二篇 选定的属性类型	3
5 选定的属性类型的定义	3
第三篇 匹配规则	21
6 串准备	21
7 匹配规则的定义	23
第四篇 上下文	37
8 上下文类型的定义	37
附录 A (规范性附录) 用 ASN.1 描述的选定的属性类型	43
附录 B (资料性附录) 属性类型概要	67
附录 C (资料性附录) 上界	69
附录 D (资料性附录) 属性、匹配规则和上下文的英文字母表顺序索引	71
附录 E (资料性附录) 区匹配规则的示例	74

国家图书馆专用

前　　言

GB/T 16264《信息技术　开放系统互连　目录》，包括以下 10 个部分：

第 1 部分：概念、模型和服务的概述；

第 2 部分：模型；

第 3 部分：抽象服务定义；

第 4 部分：分布式操作规程；

第 5 部分：协议规范；

第 6 部分：选定的属性类型；

第 7 部分：选定的客体类；

第 8 部分：公钥和属性证书框架；

第 9 部分：复制（待发布）；

第 10 部分：公用目录管理机构的系统管理用法（待发布）。

本部分是 GB/T 16264 的第 6 部分。

本部分等同采用 ISO/IEC 9594-6:2005《信息技术　开放系统互连　目录：选定的属性类型》，仅有编辑性修改。

本部分代替 GB/T 16264.6—1996。

本部分与 GB/T 16264.6—1996 的差异在于：

——增加了匹配规则；

——增加了上下文。

本部分的附录 A 是规范性附录，附录 B、附录 C、附录 D 和附录 E 是资料性附录。

本部分由中华人民共和国信息产业部提出。

本部分由全国信息技术标准化技术委员会归口。

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：徐冬梅、郑洪仁、郭楠、胡顺。

本部分于 1996 年首次发布，本次为第一次修订。

引　　言

GB/T 16264 的本部分连同本标准其他部分是为方便信息处理系统之间的互连以提供目录服务而制定的。所有这些系统的集合,连同它们所拥有的目录信息可被视为一个整体,被称为“目录”。目录所拥有的信息,总称为目录信息库(DIB),典型地被用于方便客体之间的通信、与客体的通信或有关客体的通信等,这些客体如应用实体、个人、终端和分布列表等。

目录在开放系统互连中扮演了重要角色,其目标是,在它们自身的互连标准之外做最少的技术约定的情况下,允许下述各种信息处理系统之间的互连:

- 来自不同生产厂商;
- 具有不同的管理;
- 具有不同的复杂程度,以及
- 有不同的年代。

本部分定义了一定数量的属性类型(它们在目录的一定范围的应用中可能是有用的)以及一定数量的标准属性语法和匹配规则。这里所定义的许多属性的一种特殊用途是用于名(称)格式中,尤其是用于 GB/T 16264.7—2008 中定义的客体类。

本部分提供了一个基础框架,在此框架基础上,其他标准化组织和业界论坛可以定义工业配置集。在本框架中定义为可选的许多特性,可通过配置集的说明,在某种环境下作为必选特性来使用。目前 ISO/IEC 9594 的第 5 版是原有国际标准第 4 版的修订和增强,但不是替代。在系统实现时仍可以断言为遵循第 4 版。然而,在某些方面,将不再支持第 4 版(即不再消除一些报告上的错误)。建议在系统实现时尽快遵循第 5 版。

第 5 版详细定义了目录协议的第 1 版和第 2 版。

第 1 版和第 2 版仅定义了协议第 1 版。本版(第 5 版)中定义的许多服务和协议被设计为可运行在第 1 版下。然而,一些增强的服务和协议,如署名错误,只有包含在操作中的所有的目录条目都协商支持协议第 2 版时才可运行。无论协商的是哪一版,第 5 版中所定义的服务之间的差异和协议之间的差异,除了那些特别分配给第 2 版的外,都可以使用 GB/T 16264.5—2008 中定义的扩展规则调节。

本部分使用术语“第 1 版系统”来指遵循国际标准第 1 版的所有系统,即 ISO/IEC 9594:1990 版本;本部分使用术语“第 2 版系统”来指遵循国际标准第 2 版的所有系统,即 ISO/IEC 9594:1995 版本;本部分使用术语“第 3 版系统”来指遵循国际标准第 3 版的所有系统,即 ISO/IEC 9594:1998 版本;本部分使用术语“第 4 版系统”来指遵循国际标准第 4 版的所有系统,即 ISO/IEC 9594:2001 版本的第一部分到第 10 部分;本部分使用术语“第 5 版系统”来指遵循国际标准第 5 版的所有系统,即 ISO/IEC 9594:2005 版本。

GB/T 16264—1996 是参照 ISO/IEC 9594:1990 而制定的。我国没有制定与国际标准第 2 版、第 3 版、第 4 版对应的国家标准。本部分提到的版本号是指国际标准的版本号。

附录 A 是规范性附录,提供了定义属性、属性语法和匹配规则的完整模块的 ASN.1 表示。

附录 B 是资料性附录,提供了一个属性类型表,以便于参考。

附录 C 是资料性附录,提供了本系列目录规范中建议使用的上界值。

附录 D 是资料性附录,按照字母表顺序列出了本目录规范中定义的属性和匹配规则。

附录 E 是资料性附录,给出了与区匹配定义相关的示例。

信息技术 开放系统互连 目录

第 6 部分:选定的属性类型

第一篇 概述

1 范围

GB/T 16264 的本部分定义了在各种目录应用中都很有用的若干属性类型和匹配规则。

属性类型和匹配规则可分为三类,描述如下:

某些属性类型和匹配规则可用于各种应用,或由目录本身辨别和/或使用它们。

注:在适合应用的时候,建议在生成一个新的属性类型之前,优先使用本部分中定义的属性类型或匹配规则。

某些属性类型和匹配规则在国际上已经标准化,但又是某个特殊的应用。这些属性类型和匹配规则都在与所涉及的应用有关的标准中定义。

任何管理机构可以出于任何目的规定自己的属性类型和匹配规则。它们不被国际标准化,如果创建它们的管理机构之外的机构要使用它们,仅能通过双边协商来使用。

2 规范性引用文件

下列文件中的条款通过 GB/T 16264 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 2659—2000 世界各国和地区名称代码(eqv ISO 3166-1:1997)

GB/T 9387.1—1998 信息技术 开放系统互连 基本参考模型 第 1 部分:基本模型(idt ISO/IEC 7498-1:1994)

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 16262.3—2006 信息技术 抽象语法记法一(ASN.1) 第 3 部分:约束规范(ISO/IEC 8824-3:2002, 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)

GB/T 16264.1—2008 信息技术 开放系统互连 目录 第 1 部分:概念、模型和服务的概述(ISO/IEC 9594-1:2005, IDT)

GB/T 16264.2—2008 信息技术 开放系统互连 目录 第 2 部分:模型(ISO/IEC 9594-2:2005, IDT)

GB/T 16264.3—2008 信息技术 开放系统互连 目录 第 3 部分:抽象服务定义(ISO/IEC 9594-3:2005, IDT)

GB/T 16264.4—2008 信息技术 开放系统互连 目录 第 4 部分:分布式操作规程(ISO/IEC 9594-4:2005, IDT)

GB/T 16264.5—2008 信息技术 开放系统互连 目录 第 5 部分:协议规范(ISO/IEC 9594-5:2005, IDT)

GB/T 16264.7—2008 信息技术 开放系统互连 目录 第 7 部分:选定的客体类(ISO/IEC 9594-7:2005, IDT)

ISO 639-2:1998 语言名称表示代码 第 2 部分:Alpha-3 代码

ISO/IEC 9594-8:2005 信息技术 开放系统互连 目录:公钥和属性证书框架

ISO/IEC 9594-9:2005 信息技术 开放系统互连 目录:复制

ISO/IEC 9594-10:2005 信息技术 开放系统互连 目录:公用目录管理机构的系统管理用法

ISO/IEC 9834-8:2005 信息技术 开放系统互连 OSI 登记机构的操作规程:全球唯一标识符(UUID)的产生和注册及其作为 ASN.1 客体标识符组件使用

ISO/IEC 9945-3:2003 信息技术 可移植的操作系统接口 第 3 部分:命令与实用程序

ISO/IEC 10021-4 信息技术 消息处理系统(MHS):消息传送系统 抽象服务定义和规程

ISO/IEC 10646:2003 信息技术 通用多八位编码字符集(UCS)

ITU-T 建议 E.123:2001 国家和国际电话号码、电子邮件地址和 Web 地址的记法

ITU-T 建议 E.164:2005 国际公共远程通信编号计划

ITU-T 建议 F.1:1998 国家公共电报服务的操作规定

ITU-T 建议 T.30:2005 通用交换电话网中文档传真传送规程

ITU-T 建议 T.62:1993 智能用户电报和第 4 组传真业务的控制规程

ITU-T 建议 X.121:2000 公共数据网的国际编号方案

CCITT 建议 F.31:1988 电报中继系统

CCITT 建议 F.401:1992 报文处理服务:公共报文处理服务的命名和编址

IETF RFC 3377:2002 轻量级目录访问协议(v3):技术规范

IETF RFC 3454:2002 国际化串的准备

Unicode 社团 Unicode 标准 第 4 版(Reading, MA, Addison-Wesley, 2003. ISBN 0-321-18578-1)

Unicode 标准附录 #15 Unicode 规范格式

3 术语和定义

下列术语和定义适用于 GB/T 16264 的本部分。

下列术语在 GB/T 16264.2—2008 中定义:

- a) 属性类型 *attribute type*
- b) 客体类 *object class*
- c) 匹配规则 *matching rule*
- d) 上下文 *context*

4 约定

术语“目录规范(或本目录规范)”指的是 GB/T 16264-6。术语“系列目录规范”指的是 GB/T 16264 (或者 ISO/IEC 9594)的所有部分。

本目录规范使用术语“第 1 版系统”来指遵循系列目录规范第 1 版的所有系统,即 GB/T 16264—1996 版本。本目录规范使用术语“第 2 版系统”来指遵循系列目录规范第 2 版的所有系统,即 ISO/IEC 9594:1995 版本。本目录规范使用术语“第 3 版系统”来指遵循系列目录规范第 3 版的所有系统,即 ISO/IEC 9594:1998 版本。本目录规范使用术语“第 4 版系统”来指遵循系列目录规范第 4 版的所有系统,即 ISO/IEC 9594:2001 版本的第 1 部分到第 10 部分。

本目录规范使用术语“第 5 版系统”来指遵循系列目录规范第 5 版的所有系统,即 GB/T 16264—

2008 版本的第 1 部分到第 7 部分以及 ISO/IEC 9594-8:2005、ISO/IEC 9594-9:2005 和 ISO/IEC 9594-10:2005。

本目录规范使用粗体字体来表示 ASN.1 符号。若在常规文本中要表示 ASN.1 的类型和值时,为了区别于常规文本,使用了粗体字表示。为了表示过程的语义而引用过程名时,为了区别于常规文本,使用了粗体字表示。访问控制许可使用斜体字表示。

在本部分中定义的属性类型,匹配规则和上下文类型分别使用了 GB/T 16264.2—2008 中所定义的信息客体类ATTRIBUTE、MATCHING-RULE 和CONTEXT。

对属性类型用法的示例使用一种非正式的符号来描述,在这种符号中,属性类型和值对的表示包括一个属性类型的首字母缩略语,后跟一个等于号(“=”),再后跟该属性的一个示例值。

第二篇 选定的属性类型

5 选定的属性类型的定义

本目录规范定义了一定数量的属性类型,它们在目录一定范围的应用中是有用的。

本目录规范中定义的很多属性都基于一个通用的 ASN.1 语法:

```
DirectoryString { INTEGER : maxSize } ::= CHOICE {
    teletexString          TeletexString (SIZE (1..maxSize)),
    printableString         PrintableString (SIZE (1..maxSize)),
    bmpString               BMPString (SIZE (1..maxSize)),
    universalString         UniversalString (SIZE (1..maxSize)),
    uTF8String              UTF8String (SIZE (1..maxSize)) }
```

目录的某些实现可不支持UniversalString、BMPString 或UTF8String,且因此不能产生、匹配、影像或显示具有此语法类型的属性。

5.1 系统属性类型

5.1.1 知识信息

知识信息属性类型规定了人可读的,由某个特定 DSA 所掌握的知识的累积描述。

注:这一属性目前已经废除不用。

```
knowledgeInformation ATTRIBUTE ::= {
```

WITH SYNTAX	DirectoryString { ub-knowledge-information }
EQUALITY MATCHING RULE	caseIgnoreMatch
ID	id-at-knowledgeInformation }

5.2 标记属性类型

这些属性类型与通过加标记进程与客体显式关联的客体信息相关。

5.2.1 名(称) name

名(称)属性类型是一个属性上级类型,可典型地用于构成命名的串属性类型。

```
name ATTRIBUTE ::= {
```

WITH SYNTAX	DirectoryString { ub-name }
EQUALITY MAT	CHING RULE caseIgnoreMatch
SUBSTRINGS MAT	CHING RULE caseIgnoreSubstringsMatch
ID	id-at-name }

5.2.2 公共名(称) common name

公共名(称)属性类型规定了某个客体的标识符。公共名(称)不是目录名(称);它可能是有二义性的,通过该名(称),客体在有限范围(如在一个组织内)内都知道的客体,并且符合该国的命名习俗或所

关联的文化背景。

公共名(称)的一个属性值是由它所描述的个人或组织所选择的一个串,或者对于设备和应用实体而言,是由对它所描述的客体负责的组织所选择的一个串。例如,一个英语国家的人,其典型名(称)包括一个个人头衔(如先生、女士、博士、教授、阁下、爵士)、名(称)、中间名(称)、姓、代限定符(如果有的话,如 Jr.)以及勋章和奖励(如果有的话,如 QC)等。

举例:

```
CN="Mr. Robin Lachlan McLeod BSc(Hons) CEng MIEE";
CN="Divisional Coordination Committee";
CN="High Speed Modem".
```

任何变量都应作为独立的和可替代的属性值与被命名的客体关联起来。

其他一些通用的变量也应允许,如将中间名(称)作为首选的名(称)使用;使用“Bill”代替“William”,等等。

```
commonName ATTRIBUTE ::= {
```

```
    SUBTYPE OF      name
    WITH SYNTAX    DirectoryString {ub-common-name}
    ID             id-at-commonName }
```

5.2.3 姓 surname

姓属性类型指定某个个体从其父亲或配偶继承而来的语言结构,通过姓,该个体被人们普遍知晓。

姓的一个属性值是一个串,如“McLeod”。

```
surname ATTRIBUTE ::= {
```

```
    SUBTYPE OF      name
    WITH SYNTAX    DirectoryString {ub-surname}
    ID             id-at-surname }
```

5.2.4 给定名 given name

给定名属性类型规定了一种语言学结构,它通常由某个个体的父母为该个体给出,或者由个体本人选择,或者通过该名(称),该个体被人们普遍知晓。

给定名的一个属性值是一个串,如“David”或“Jean Paul”。

```
givenName ATTRIBUTE ::= {
```

```
    SUBTYPE OF      name
    WITH SYNTAX    DirectoryString {ub-name}
    ID             id-at-givenName }
```

5.2.5 首字母 initials

首字母属性类型包含某个个体名(称)的除姓以外的其他某些或全部名(称)的首字母。

首字母的一个属性值是一个串,如“D”或“D.”或“J. P.”。

```
initials ATTRIBUTE ::= {
```

```
    SUBTYPE OF      name
    WITH SYNTAX    DirectoryString {ub-name}
    ID             id-at-initials }
```

5.2.6 代限定符 generation qualifier

代限定符属性类型包含一个用于提供“代”信息的串,以便限定该个体的名(称)。

代限定符的一个属性值是一个串,如“Jr.”或“II”。

```
generationQualifier ATTRIBUTE ::= {
```

```
    SUBTYPE OF      name
```

WITH SYNTAX	DirectoryString {ub-name}
ID	id-at-generationQualifier }

5.2.7 唯一标识符 **unique identifier**

唯一标识符属性类型规定了一个标识符,当某个辨别名被重复使用时,该标识符可用来区分客体引用。例如,它可能是一个编码后的客体标识符、证书、日期、时戳或可证明该辨别名有效性的其他一些形式。

唯一标识符的一个属性值是一个比特串。

uniqueIdentifier ATTRIBUTE ::= {

WITH SYNTAX	UniqueIdentifier
EQUALITY MATCHING RULE	bitStringMatch
ID	id-at-uniqueIdentifier }

UniqueIdentifier ::= BIT STRING

5.2.8 可辨别名限定符 **DN qualifier**

可辨别名限定符属性类型规定了要增加到某个条目的相对辨别名中的无二义性信息。它将被用于在多个DSA中都拥有的那些条目中,如果不增加该限定符,则这些条目将具有相同的名(称),并且在某个给定的DSA内,对于要加入该信息的所有条目而言,该限定符的值是相同的。

dnQualifier ATTRIBUTE ::= {

WITH SYNTAX	PrintableString
EQUALITY MATCHING RULE	caseIgnoreMatch
ORDERING MATCHING RULE	caseIgnoreOrderingMatch
SUBSTRINGS MATCHING RULE	caseIgnoreSubstringsMatch
ID	id-at-dnQualifier }

5.2.9 序列号 **serial number**

序列号属性类型规定了某个客体的序列号的标识符。

序列号的一个属性值是一个可打印的串。

serialNumber ATTRIBUTE ::= {

WITH SYNTAX	PrintableString (SIZE (1.. ub-serial-number))
EQUALITY MATCHING RULE	caseIgnoreMatch
SUBSTRINGS MATCHING RULE	caseIgnoreSubstringsMatch
ID	id-at-serialNumber }

5.2.10 假名 **pseudonym**

假名属性类型为某个客体规定了一个假名。如果已经很明确客体的名(称)是一个假名时,则使用假名来对客体进行命名。

pseudonym ATTRIBUTE ::= {

SUBTYPE OF	name
WITH SYNTAX	DirectoryString {ub-pseudonym}
ID	id-at-pseudonym }

5.2.11 全球唯一标识符对 **universal unique identifier pair**

全球唯一标识符对属性类型规定了一对全球唯一标识符(UUID),如同在ISO/IEC 9834-8中规定的那样。该对共同表示了一个发起者/服从者关系,这种关系的特性不在本目录规范的定义范围之内。在该对中,初始UUID表示发起者,而该对中后面的UUID表示发起者/服从者关系中的服从者。这种关系的一个示例为一个用户账号。

UUIDPair ATTRIBUTE ::= {

```

WITH SYNTAX          UUIDPair
EQUALITY MATCHING RULE   UUIDPairMatch
ID                  id-at-uuidpair }

UUIDPair ::= SEQUENCE {
    issuerUUID        UUID,
    subjectUUID       UUID }

```

UUID ::= OCTET STRING (SIZE(16))-- 仅使用 UUID 格式

5.3 地理属性类型

这些属性类型涉及有关客体所处的地理位置或区域。

5.3.1 国家名 country name

国家名属性类型规定了一个国家。如果作为一个目录名(称)的组成部分,则它表示被命名的客体在物理上所处的国家,或者在其他某些重要方面客体与之相关联的国家。

国家名的一个属性值选自 GB/T 2659—2000 的一个串。

countryName ATTRIBUTE ::= {

```

SUBTYPE OF      name
WITH SYNTAX    CountryName
SINGLE VALUE   TRUE
ID             id-at-countryName }

```

CountryName ::= PrintableString (SIZE(2))-- 仅使用 GB/T 2659—2000 代码

5.3.2 地点名 locality name

地点名属性类型规定了一个地点。当作为目录名(称)的一部分使用时,则它标识被命名的客体实际所处的地理地区或地理位置,或者在其他某些重要方面客体与之相关联的地理地区或地点。

地点名的属性值是一个串,如 L=“Edinburgh”。

localityName ATTRIBUTE ::= {

```

SUBTYPE OF      name
WITH SYNTAX    DirectoryString {ub-locality-name}
ID             id-at-localityName }

```

集合地点名属性类型为条目集合规定了一个地点名。

collectiveLocalityName ATTRIBUTE ::= {

```

SUBTYPE OF      localityName
COLLECTIVE    TRUE
ID             id-at-collectiveLocalityName }

```

5.3.3 州或省名 state or province name

省或州名属性类型规定了一个州或一个省。如果作为一个目录名(称)的组成部分,则它表示被命名的客体在物理上所处的一个地理分区,或者在其他某些重要方面客体与之相关联的一个地理分区。

省或州名的一个属性值是一个串,如 S=“Ohio”。

stateOrProvinceName ATTRIBUTE ::= {

```

SUBTYPE OF      name
WITH SYNTAX    DirectoryString {ub-state-name}
ID             id-at-stateOrProvinceName }

```

集合省或州名属性类型为条目集合规定了一个省或州名。

collectiveStateOrProvinceName ATTRIBUTE ::= {

```

SUBTYPE OF      stateOrProvinceName

```

COLLECTIVE	TRUE
ID	id-at-collectiveStateOrProvinceName }

5.3.4 街道地址 street address

街道地址属性类型规定一个本地分发或实际投递的邮政地址,即街道名(称)、场所、大街和门牌号码等。如果作为一个目录名(称)的组成部分,则它标识被命名的客体所处的街道地址,或者通过其他方式与之建立联系的街道地址。

街道地址的属性值是一个串,如“Changanjie 888”。

```
streetAddress ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-street-address}
    EQUALITY MATCHING RULE  caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID                  id-at-streetAddress }
```

集合街道地址属性类型为条目集合规定了一个街道地址。

```
collectiveStreetAddress ATTRIBUTE ::= {
    SUBTYPE OF           streetAddress
    COLLECTIVE          TRUE
    ID                  id-at-collectiveStreetAddress }
```

5.3.5 住宅标识符 house identifier

住宅标识符属性类型规定了用于标识某个具体建筑物的语言学结构,例如一个住宅号码或与街道、大街、城镇或城市等相关的住宅名(称)。

住宅标识符的属性值是一个串,如“14”。

```
houseIdentifier ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-name}
    EQUALITY MATCHING RULE  caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID                  id-at-houseIdentifier }
```

5.4 组织属性类型

这些属性类型与组织相关,并且可用于描述与这些组织有关的客体。

5.4.1 组织名 organization name

组织名属性类型规定了一个组织。当作为目录名的一部分使用时,它标识命名客体实际所属的组织。

OrganizationName 的属性值是由组织所选定的串(例如:O = "Scottish Telecommunicationsplc")。任何变量都应作为独立的和可替代的属性值与被命名的组织相关联起来。

```
organizationName ATTRIBUTE ::= {
    SUBTYPE OF           name
    WITH SYNTAX          DirectoryString {ub-organization-name}
    ID                  id-at-organizationName }
```

集合组织名属性类型为条目集合规定了组织名。

```
collectiveOrganizationName ATTRIBUTE ::= {
    SUBTYPE OF           organizationName
    COLLECTIVE          TRUE
    ID                  id-at-collectiveOrganizationName }
```

5.4.2 组织单元名 organizational unit name

组织单元名属性类型规定了一个组织单元。当作为一个目录名(称)的组成部分,它标识命名客体实际所属的组织单元。

指定的组织单元被理解为是由属性organizationName 所指定的组织中的一部分。因此,如果组织单元名被用于某个目录名(称)中,则它必须与一个organizationName 属性相关联起来使用。

组织单元名的属性值是由该组织单元所在的组织所选定的串(如 OU="Technology Division")。注意,通常使用的缩写“TD”应当作为一个独立的和可替代的属性值。

示例:

```
O="Scottel",OU="TD"
organizationalUnitName ATTRIBUTE ::= {
    SUBTYPE OF           name
    WITH SYNTAX          DirectoryString { ub-organizational-unit-name }
    ID                   id-at-organizationalUnitName }
```

集合组织单元名属性类型为条目集合规定了组织单元名。

```
collectiveOrganizationalUnitName ATTRIBUTE ::= {
    SUBTYPE OF           organizationalUnitName
    COLLECTIVE          TRUE
    ID                  id-at-collectiveOrganizationalUnitName }
```

5.4.3 头衔 title

头衔属性类型规定了客体在某个组织内被指定的地位或功能。

头衔属性值为串。

示例:

```
T="Manager,Distributed Applications"
title ATTRIBUTE ::= {
    SUBTYPE OF           name
    WITH SYNTAX          DirectoryString { ub-title }
    ID                   id-at-title }
```

5.5 解释属性类型

这些属性类型与客体的某些情况的解释相关(如以自然语言描述)。

5.5.1 描述 description

描述属性类型规定了描述相关客体的文本。

例如,客体“Standards Interest”可以具有如下描述:“用于交换公司内部标准开发的信息分发表”。

描述的属性值是一个串。

```
description ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString { ub-description }
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID                   id-at-description }
```

5.5.2 搜索指南 search guide

搜索指南属性类型规定了所建议的搜索准则信息,这些信息可包含在某些可作为搜索操作的基客体的条目中,例如国家或组织。

搜索准则包含客体搜索类型的可选客体标识符以及构造过滤器的属性类型和逻辑操作符的组合。也可以为每个搜索准则项指定匹配级别,如近似匹配。

搜索指南属性可重复出现以反映各种不同类型的请求,例如,对某个住宅个人或组织个人的搜索,可能会从给定的基客体开始通过阅读搜索指南来实现。

```

searchGuide ATTRIBUTE ::= {
    WITH SYNTAX      Guide
    ID               id-at-searchGuide }

Guide ::= SET {
    objectClass [0]      OBJECT-CLASS. &.id OPTIONAL,
    criteria   [1]      Criteria }

Criteria ::= CHOICE {
    Type [0]  CriteriaItem,
    and    [1]  SET OF Criteria,
    or     [2]  SET OF Criteria,
    not    [3]  Criteria }

CriteriaItem ::= CHOICE {
    equality      [0]  AttributeType,
    substrings    [1]  AttributeType,
    greaterOrEqual [2]  AttributeType,
    lessOrEqual   [3]  AttributeType,
    approximateMatch [4] AttributeType }

```

示例:

下述内容是搜索指南属性的一个潜在取值,可被存储于客体类为 Locality 的条目中,以便指示客体类为 ResidentialPerson 的条目是如何被发现的:

```

residential-person-guide Guide ::= {
    objectClass residentialPerson. &.id,
    criteria and : {
        type : substrings : commonName. &.id,
        type : substrings : streetAddress. &.id } }

```

根据此指南值来构造一个过滤器是很直接的。

第(1)步,产生一个中间过滤器值:

```

intermediate-filter Filter ::= {
    and : {
        item : substrings {
            type commonName. &.id,
            strings { any : teletexString : "Dubois" } },
        item : substrings {
            type streetAddress. &.id,
            strings { any : teletexString "Hugo" } } } }

```

第(2)步,产生一个用于在子树中匹配住宅个人条目的过滤器:

```

residential-person-filter Filter ::= {
    and : {
        item : equality : {
            type objectClass. &.id,
            assertion residentialPerson. &.id } } }

```

```
intermediateFilter }
```

5.5.3 增强的搜索指南 enhanced search guide

增强的搜索指南属性提供了对searchGuide 属性的一种增强,为了能够在某个给定客体类的下级客体中进行搜索,增加了建议的搜索深度信息。

```
enhancedSearchGuide ATTRIBUTE ::= {
```

WITH SYNTAX	EnhancedGuide
ID	id-at-enhancedSearchGuide }

```
EnhancedGuide ::= SEQUENCE {
```

objectClass	[0] OBJECT-CLASS. &.id,
criteria	[1] Criteria,
subset	[2] INTEGER

```
{ baseObject (0), oneLevel (1), wholeSubtree (2) } DEFAULT oneLevel }
```

5.5.4 商业类别 business category

商业类别属性类型规定了与某些通用客体(如人)的职业相关的信息。例如,该属性提供了一种方式可以查询目录以获取同行业人员的信息。

```
businessCategory ATTRIBUTE ::= {
```

WITH SYNTAX	DirectoryString {ub-business-category}
EQUALITY MATCHING RULE	caseIgnoreMatch
SUBSTRINGS MATCHING RULE	caseIgnoreSubstringsMatch
ID	id-at-businessCategory }

5.6 邮政编址属性类型

这些属性类型涉及对某个客体进行物理邮政投递时所需的信息。

5.6.1 邮政地址 postal address

邮政地址属性类型规定命名客体的邮政管理机构实际投递一个邮件所要求的地址信息。

邮政地址的属性值由根据 CCITT F. 401 建议中定义的 MHS 无格式邮政 O/R 地址版本 1 所选择的属性构成,并且地址长度为 6 行,每行最多 30 个字符,其中包括一个投递国家名。在通常情况下,根据命名客体的特殊要求,这种地址包括以下几个部分,即:邮件的接收者名、街道地址、城市名、省或州名、邮政编码和邮政信箱号。

```
postalAddress ATTRIBUTE ::= {
```

WITH SYNTAX	PostalAddress
EQUALITY MATCHING RULE	caseIgnoreListMatch
SUBSTRINGS MATCHING RULE	caseIgnoreListSubstringsMatch
ID	id-at-postalAddress }

```
PostalAddress ::= SEQUENCE SIZE(1.. ub-postal-line) OF DirectoryString {ub-postal-string}
```

集合邮政地址属性类型为条目集合指定了一个邮政地址。

```
collectivePostalAddress ATTRIBUTE ::= {
```

SUBTYPE OF	postalAddress
COLLECTIVE	TRUE
ID	id-at-collectivePostalAddress }

5.6.2 邮政编码 postal code

邮政编码属性类型规定已命名客体的邮政编码。如果该属性值存在,则它应当是客体邮政地址中的组成部分。

邮政编码的属性值是一个串。

```

postalCode ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString { ub-postal-code }
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID                   id-at-postalCode }

```

集合邮政编码属性类型为条目的集合规定了一个邮政编码。

```

collectivePostalCode ATTRIBUTE ::= {
    SUBTYPE OF           postalCode
    COLLECTIVE          TRUE
    ID                  id-at-collectivePostalCode }

```

5.6.3 邮政信箱 post office box

邮政信箱属性类型规定了客体可以收到邮政投递的邮政信箱。如果该属性值存在，则它应当是客体邮政地址中的组成部分。

```

postOfficeBox ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString { ub-post-office-box }
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID                   id-at-postOfficeBox }

```

集合邮政信箱属性类型为条目集合规定了一个邮政信箱。

```

collectivePostOfficeBox ATTRIBUTE ::= {
    SUBTYPE OF           postOfficeBox
    COLLECTIVE          TRUE
    ID                  id-at-collectivePostOfficeBox }

```

5.6.4 物理投递邮局名 physical delivery office name

物理投递邮局名属性类型规定了某个物理投递邮局所在的城市、乡村等的名(称)。

物理投递邮局名的属性值是一个串。

```

physicalDeliveryOfficeName ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString { ub-physical-office-name }
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID                   id-at-physicalDeliveryOfficeName }

```

集合物理投递邮局名属性类型为条目集合规定了一个物理投递邮局名。

```

collectivePhysicalDeliveryOfficeName ATTRIBUTE ::= {
    SUBTYPE OF           physicalDeliveryOfficeName
    COLLECTIVE          TRUE
    ID                  id-at-collectivePhysicalDeliveryOfficeName }

```

5.7 远程通信编址属性类型

这些属性类型涉及使用远程通信方式与客体进行通信时所需的编址信息。

5.7.1 电话号码 telephone number

电话号码属性类型规定了与某个客体相关的一个电话号码。

电话号码的属性值是一个符合国际约定格式(ITU-T E.123 建议)的国际电话号码串(如“+44 582 10101”)。

```
telephoneNumber ATTRIBUTE ::= {
```

```

WITH SYNTAX           TelephoneNumber
EQUALITY MATCHING RULE   telephoneNumberMatch
SUBSTRINGS MATCHING RULE   telephoneNumberSubstringsMatch
ID                   id-at-telephoneNumber ber }

TelephoneNumber ::= PrintableString (SIZE(1.. ub-telephone-number))
-- 仅使用符合 ITU-T E. 123 建议书中规定的串
集合电话号码属性类型为条目集合规定了一个电话号码。
collectiveTelephoneNumber ATTRIBUTE ::= {
  SUBTYPE OF           telephoneNumber
  COLLECTIVE          TRUE
  ID                  id-at-collectiveTelephoneNumber }

```

5.7.2 电报号码 telex number

电报号码属性类型规定了与某个客体相关联的电报终端的号码、国家码和回答码。

```

telexNumber ATTRIBUTE ::= {
  WITH SYNTAX           TelexNumber
  ID                   id-at-telexNumber }

TelexNumber ::= SEQUENCE {
  telexNumber           PrintableString (SIZE (1.. ub-telex-number)),
  countryCode          PrintableString (SIZE (1.. ub-country-code)),
  answerback           PrintableString (SIZE (1.. ub-answerback)) }

集合电报号码属性类型为条目集合规定了一个电报号码。
collectiveTelexNumber ATTRIBUTE ::= {
  SUBTYPE OF           telexNumber
  COLLECTIVE          TRUE
  ID                  id-at-collectiveTelexNumber }

```

5.7.3 智能用户电报终端标识符 teletex terminal identifier

自从 CCITT F. 200 建议书被取消后,目前尚无替代建议书,属性类型 *teletexTerminalIdentifier* 和 *collectiveTeletexTerminalIdentifier* 的用法被反对。

智能用户电报终端标识符属性类型规定一个与某个客体相关联的智能用户电报终端的标识符(以及可选的参数)。

智能用户电报终端标识符的属性值为一个串,符合 CCITT F. 200 建议书和一个任选集,该任选集中的组件符合 ITU-T T. 62 建议书。

```

-- teletexTerminalIdentifier ATTRIBUTE ::= {
--   WITH SYNTAX      TeletexTerminalIdentifier
--   ID              id-at-teletexTerminalIdentifier }

TeletexTerminalIdentifier ::= SEQUENCE {
  teletexTerminal     PrintableString (SIZE(1.. ub-teletex-terminal-id)),
  parameters         TeletexNonBasicParameters OPTIONAL }

集合智能用户电报终端标识符属性类型为条目集合指定了一个智能用户电报终端标识符。

```

```

-- collectiveTeletexTerminalIdentifier ATTRIBUTE ::= {
--   SUBTYPE OF      teletexTerminalIdentifier
--   COLLECTIVE      TRUE
--   ID              id-at-collectiveTeletexTerminalIdentifier }

```

5.7.4 传真电话号码 facsimile telephone number

传真电话号码属性类型为与某个客体相关联的传真终端指定了一个电话号码(以及可选的参数)。

传真电话号码的属性值是一个串,符合 ITU-T E.123 建议书中表示国际电话号码的国际协定格式(如“+81 3 347 7418”)以及一个可选的比特串(格式符合 ITU-T T.30 建议书)。

facsimileTelephoneNumber ATTRIBUTE ::= {

WITH SYNTAX	FacsimileTelephoneNumber
EQUALITY MATCHING RULE	facsimileNumberMatch
SUBSTRINGS MATCHING RULE	facsimileNumberSubstringsMatch
ID	id-at-facsimileTelephoneNumber
FacsimileTelephoneNumber ::= SEQUENCE {	
telephoneNumber	TelephoneNumber,
parameters	G3FacsimileNonBasicParameters OPTIONAL

集合传真电话号码属性类型为条目集合指定了一个传真电话。

collectiveFacsimileTelephoneNumber ATTRIBUTE ::= {

SUBTYPE OF	facsimileTelephoneNumber
COLLECTIVE	TRUE
ID	id-at-collectiveFacsimileTelephoneNumber }

5.7.5 X.121 地址 X.121 address

X.121 地址属性类型规定了一个与某客体相关联的、按照 ITU-T X.121 建议书定义的地址。

X121Address ATTRIBUTE ::= {

WITH SYNTAX	X121Address
EQUALITY MATCHING RULE	numericStringMatch
SUBSTRINGS MATCHING RULE	numericStringSubstringsMatch
ID	id-at-x121Address }

X121Address ::= NumericString (SIZE(1.. ub-x121-address))

-- ITU-T X.121 建议书中定义的串

5.7.6 国际 ISDN 号码 international ISDN number

国际 ISDN 号码属性类型规定了一个与某客体相关联的国际 ISDN 号码。

国际 ISDN 号码的属性值是一个串,符合 ITU-T E.164 建议书给出的 ISDN 地址的国际协定格式。

internationalISDNNumber ATTRIBUTE ::= {

WITH SYNTAX	InternationalISDNNumber
EQUALITY MATCHING RULE	numericStringMatch
SUBSTRINGS MATCHING RULE	numericStringSubstringsMatch
ID	id-at-internationalISDNNumber }

InternationalISDNNumber ::= NumericString (SIZE(1.. ub-international-isdn-number))

-- 仅使用符合 ITU-T E.164 建议书中规定的串

集合国际 ISDN 号码属性类型为条目集合规定了一个国际 ISDN 号码。

collectiveInternationalISDNNumber ATTRIBUTE ::= {

SUBTYPE OF	internationalISDNNumber
COLLECTIVE	TRUE
ID	id-at-collectiveInternationalISDNNumber }

5.7.7 注册地址 registered address

注册地址属性类型规定了一个与某个客体相关联的位于特定城市位置中的地址助记符。该助记符

在城市所在的国家进行了注册,并且按照公用电报业务的规定来使用(根据 ITU-T F.1 建议书)。

```
registeredAddress ATTRIBUTE ::= {
    SUBTYPE OF      postalAddress
    WITH SYNTAX    PostalAddress
    ID              id-at-registeredAddress }
```

5.7.8 目的地指示符 destination indicator

目的地指示符属性类型规定了与某个客体(收信人)相关联的需要提供公用电报业务的国家和城市(根据 ITU-T F.1 建议书和 CCITT F.31 建议书)。

目的地指示符的属性值是一个串。

```
destinationIndicator ATTRIBUTE ::= {
    WITH SYNTAX          DestinationIndicator
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID                   id-at-destinationIndicator or }
```

DestinationIndicator ::= PrintableString (SIZE(1.. ub-destination-indicator))

-- 仅使用字母表字符

5.7.9 通信业务 communications service

通信业务属性类型规定了与某个通信地址相关联的业务类型。

```
communicationsService ATTRIBUTE ::= {
    WITH SYNTAX          CommunicationsService
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                   id-at-communicationsService }
```

CommunicationsService ::= OBJECT IDENTIFIER

该属性描述了通信地址可以提供访问的业务的种类,例如电话(话音)、传真、电子邮件、SMS(短消息业务)、EDI 和文件传输等。

为标识业务而进行的客体标识符的分配不在本目录规范的定义范围之内。

5.7.10 通信网络 communications network

通信网络属性类型规定了某个通信地址所使用的网络类型。

```
communicationsNetwork ATTRIBUTE ::= {
    WITH SYNTAX          CommunicationsNetwork
    EQUALITY MATCHING RULE objectIdentifierMatch
    SINGLE VALUE          TRUE
    ID                   id-at-communicationsNetwork }
```

CommunicationsNetwork ::= OBJECT IDENTIFIER

该属性描述了通信地址所处的网络的类型。例如,一个公众交换电话网(PSTN)、一个 ISDN 网或一个 GSM 移动电话网等。它还可以是一个面向应用的网络,如一个银行网。

为标识网络而进行的客体标识符的分配不在本目录规范的定义范围之内。

5.8 优选的属性类型

这些属性类型涉及客体的优选。

5.8.1 优选投递方式 preferred delivery method

优选投递方式属性类型规定了在与客体通信时所使用的方式的优先权顺序。

```
preferredDeliveryMethod ATTRIBUTE ::= {
```

WITH SYNTAX	PreferredDeliveryMethod
-------------	-------------------------

SINGLE VALUE	TRUE
ID	id-at-preferredDeliveryMethod }

PreferredDeliveryMethod ::= SEQUENCE OF INTEGER {

any-delivery-method	(0),
mhs-delivery	(1),
physical-delivery	(2),
telex-delivery	(3),
teletex-delivery	(4),
g3-facsimile-delivery	(5),
g4-facsimile-delivery	(6),
ia5-terminal-delivery	(7),
videotex-delivery	(8),
telephone-delivery	(9) }

5.9 OSI 应用属性类型

这些属性类型与 OSI 应用层中的客体所涉及的信息相关。

5.9.1 表示地址 presentation address

表示地址属性类型规定了与表示 OSI 应用实体的某个客体相关联的表示地址。

表示地址的属性值是在 GB/T 9387.1—1998 中定义的一个表示地址。

presentationAddress ATTRIBUTE ::= {	
WITH SYNTAX	PresentationAddress
EQUALITY MATCHING RULE	presentationAddressMatch
SINGLE VALUE	TRUE
ID	id-at-presentationAddress }

PresentationAddress ::= SEQUENCE {

pSelector	[0] OCTET STRING OPTIONAL,
sSelector	[1] OCTET STRING OPTIONAL,
tSelector	[2] OCTET STRING OPTIONAL,
nAddresses	[3] SET SIZE (1.. MAX) OF OCTET STRING }

5.9.2 支持的应用上下文 supported application context

支持的应用上下文属性类型规定了某个客体(一个 OSI 应用实体)所支持的应用上下文的客体标识符。

supportedApplicationContext ATTRIBUTE ::= {	
WITH SYNTAX	OBJECT IDENTIFIER
EQUALITY MATCHING RULE	objectIdentifierMatch
ID	id-at-supportedApplicationContext }

5.9.3 协议信息 protocol information

协议信息属性类型将协议信息与表示地址属性中的每个网络地址相关联。

对于每一个nAddress,协议组件都标识了网络层和传输层的协议或概要。

protocolInformation ATTRIBUTE ::= {	
WITH SYNTAX	ProtocolInformation
EQUALITY MATCHING RULE	protocolInformationMatch
ID	id-at-protocolInformation }

ProtocolInformation ::= SEQUENCE {

nAddress	OCTET STRING,
profiles	SET OF OBJECT IDENTIFIER }

5.10 关系属性类型

这些属性类型涉及以某种方式与特定客体有关的客体信息。

注：在这些属性类型中使用的语法DistinguishedName 允许使用主辨别名或一个替代辨别名。在主辨别名已知的情况下，如果使用主辨别名，可以确保与 1997 年前的 DSA 保持一致性和互操作性。某些特定的用法可能会要求使用某个特殊的替代名(称)。上下文信息和可替代辨别值也可能作为任何 RDN 中组件valuesWithContext 的一部分，如在 GB/T 16264.2—2008 的 9.3 所描述的。

5.10.1 可辨别名 distinguished name

可辨别名属性类型是规定一个客体名(称)的属性。

distinguishedName ATTRIBUTE ::= {

WITH SYNTAX	DistinguishedName
EQUALITY MATCHING RULE	distinguishedNameMatch
ID	id-at-distinguishedName }

5.10.2 成员 member

成员属性类型规定了与某个客体相关联的一组名(称)。

成员的一个属性值是一个辨别名。

member ATTRIBUTE ::= {

SUBTYPE OF	distinguishedName
ID	id-at-member

5.10.3 唯一成员 unique member

唯一成员属性类型规定了与某个客体相关联的一组唯一名(称)。唯一名(称)指的是通过可选地包含其唯一标识符而无二义性的一个名(称)。

唯一成员的属性值是一个辨别名，并带有一个可选的唯一标识符。

uniqueMember ATTRIBUTE ::= {

WITH SYNTAX	NameAndOptionalUID
EQUALITY MATCHING RULE	uniqueMemberMatch
ID	id-at-uniqueMember }

NameAndOptionalUID ::= SEQUENCE {

dn	DistinguishedName,
uid	UniqueIdentifier OPTIONAL }

5.10.4 所有者 owner

所有者属性类型规定了对相关联的客体负有责任的某个客体的名(称)。

所有者的属性值是一个辨别名(可以表示一组名(称))，并且能够重复出现。

owner ATTRIBUTE ::= {

SUBTYPE OF	distinguishedName
ID	id-at-owner }

5.10.5 角色承担者 role occupant

角色承担者属性类型规定了某个客体的名(称)，该客体承担了组织内的某个角色。

角色承担者的属性值是一个辨别名。

RoleOccupant ATTRIBUTE ::= {

SUBTYPE OF	distinguishedName
ID	id-at-roleOccupant }

5.10.6 另见 see also

另见属性类型规定了其他目录客体的名(称),这些目录客体可能是同一个现实世界客体在其他方面(在某种意义上说)的体现。

另见的属性值是一个辨别名。

`seeAlso ATTRIBUTE ::= {`

SUBTYPE OF	distinguishedName
ID	id-at-seeAlso }

5.11 域属性类型

5.11.1 DMD 名 DMD name

DMD 名属性类型规定了一个目录管理域(DMD)。如果作为一个目录名(称)中的组成部分,则它标识了管理已命名客体的一个目录管理域。

DMD 名的属性值是由 DMD 所选择的一个串。

`dmdName ATTRIBUTE ::= {`

SUBTYPE OF	name
WITH SYNTAX	DirectoryString{ub-common-name}
ID	id-at-dmdName }

5.12 通知属性

通知属性具有属性语法,但是是用来在元素 CommonResults (或 CommonResultsSeq) 和 Partial OutcomeQualifier 中承载附加信息(正如在 GB/T 16264.6—2008 的 7.4 和 10.1 所描述的那样)。它们经常与匹配规则一起定义,因此可以检查返回的值是否与本地已知值相匹配。

5.12.1 DSA 问题 DSA problem

DSA 问题通知属性与一个 serviceError 或一个 PartialOutcomeQualifier 同时使用,定义如下:

`dSAPrblem ATTRIBUTE ::= {`

WITH SYNTAX	OBJECT IDENTIFIER
EQUALITY MATCHING RULE	objectIdentifierMatch
ID	id-not-dSAPrblem }

为 dsaProblem 定义的值包括:

- id-pr-targetDsaUnavailable: 在名(称)解析中,请求被链接至另一个 DSA,但是与该目标 DSA 的连接无法建立;
- id-pr-dataSourceUnavailable: DSA 不能完成该操作,因为 DIB 的部分不可用;
- id-pr-administratorImposedLimit: 某个操作已经超出了管理者所设置的某些限制;
- id-pr-permanentRestriction: 某个操作使得 DSA 超出了某些限制,使得进程停止,且重复的操作被判断为仍然会碰到同样的问题;
- id-pr-temporaryRestriction: 某个操作使得 DSA 超出了某些限制,使得进程停止,但是问题被判断为是一个临时问题,如由于资源损耗。

5.12.2 搜索服务问题 search service problem

搜索服务问题通知属性描述了在应用搜索规则策略时所遇到的问题,它与 service-errors 或 PartialOutcomeQualifier 共同使用。定义如下:

`searchServiceProblem ATTRIBUTE ::= {`

WITH SYNTAX	OBJECT IDENTIFIER
EQUALITY MATCHING RULE	objectIdentifierMatch
SINGLE VALUE	TRUE
ID	id-not-searchServiceProblem }

为 searchServiceProblem 定义的值包括：

- a) id-pr-unidentifiedOperation: 试图执行的操作不符合该服务所标识的那些操作之一；
- b) id-pr-unavailableOperation: 试图执行的操作仅符合一个对于请求者而言不可用的搜索规则；
- c) id-pr-searchAttributeViolation: 要求在过滤器中出现的一个或多个属性类型没有出现；
- d) id-pr-searchAttributeCombinationViolation—search 请求的过滤器中没有包括所要求的属性类型的组合；
- e) id-pr-searchValueNotAllowed: 为某些属性类型指定了属性值，但仅仅是在 present 和 context-Present 过滤项类型中的属性类型才可以被指定；
- f) id-pr-missingSearchAttribute: 在相关的搜索规则中要求出现被标识的属性，但该属性并没有出现在所请求的搜索中；
- g) id-pr-searchValueViolation: 在使用相关的搜索规则进行搜索时，所标识的指定属性类型的属性值是不允许的；
- h) id-pr-attributeNegationViolation: 在搜索过滤器中，所标识的属性类型是不允许出现在否定形式中的；
- i) id-pr-searchValueRequired: 所标识的属性类型是不允许出现在不要求值匹配的过滤项中的；
- j) id-pr-invalidSearchValue: 对于相关的搜索规则，所标识的属性值对于指定的属性类型而言是无效的；
- k) id-pr-searchContextViolation: 在试图进行的搜索中，所标识的上下文类型对于属性类型而言是不允许的；
- l) id-pr-searchContextCombinationViolation: 在相关的搜索规则中要求出现所标识的上下文类型的组合，但该组合并没有出现在所请求的搜索中；
- m) id-pr-missingSearchContext: 属性类型要求出现所标识的上下文类型，但该类型并没有出现在所请求的搜索中；
- n) id-pr-searchContextValueViolation: 所标识的上下文类型的上下文值，对于属性类型而言是不允许的；
- o) id-pr-searchContextValueRequired: 所标识的属性类型是不允许出现在不要求值匹配的过滤项中的；
- p) id-pr-invalidContextSearchValue: 对于相关的搜索规则，所标识的属性值对于指定的属性类型而言是无效的；
- q) id-pr-unsupportedMatchingRule: 不支持所标识的请求匹配规则；
- r) id-pr-attributeMatchingViolation: 对于相关的搜索规则，所标识的请求匹配规则或其具体用法对于指定的属性而言是不允许的；
- s) id-pr-unsupportedMatchingUse: 不支持某个匹配规则在搜索过滤器中的建议用法；
- t) id-pr-matchingUseViolation: 不允许某个匹配规则在搜索过滤器中的建议用法，例如在搜索规则中指定的用法；
- u) id-pr-hierarchySelectForbidden: 对于指定类型的请求，不允许进行除了 self 之外的层次选择；
- v) id-pr-invalidHierarchySelect: 在请求中指定了一个或多个无效的层次选择选项；
- w) id-pr-unavailableHierarchySelect: 在实现时不支持一个或多个层次选择；
- x) id-pr-invalidSearchControlOptions: 在请求中指定了一个或多个无效的搜索选项；
- y) id-pr-invalidServiceControlOptions: 在请求中指定了一个或多个无效的服务控制选项；
- z) id-pr-searchSubsetViolation: 对于相关的搜索规则而言，不允许所请求的搜索子集；
- aa) id-pr-unmatchedKeyAttributes: 选择了一个基于映射的匹配规则，但可映射的过滤项不提供任何针对相关映射表的匹配；

- bb) id-pr-ambiguousKeyAttributes: 选择了一个基于映射的匹配规则,但可映射的过滤项提供了多个针对相关映射表的匹配;
- cc) id-pr-unavailableRelaxationLevel: DSA 不支持某个所请求的放宽扩展级别;
- dd) id-pr-emptyHierarchySelection: 指定了一个层次选择但结果是没有一个条目返回,尽管有一个或多个条目是与搜索过滤器相匹配的;
- ee) id-pr-relaxationNotSupported: 在用户请求中指定了放宽,但该放宽不被支持。

5.12.3 服务类型 service type

服务类型通知属性给出失败搜索的服务类型。

serviceType ATTRIBUTE ::= {

WITH SYNTAX	OBJECT IDENTIFIER
EQUALITY MATCHING RULE	objectIdentifierMatch
SINGLE VALUE	TRUE
ID	id-not-serviceType }

5.12.4 属性类型列表 attribute type list

属性类型列表通知属性给出了一个属性类型的列表,以便更进一步限定搜索服务的问题。

attributeTypeList ATTRIBUTE ::= {

WITH SYNTAX	OBJECT IDENTIFIER
EQUALITY MATCHING RULE	objectIdentifierMatch
ID	id-not-attributeTypeList }

5.12.5 匹配规则列表 matching rule list

匹配规则列表通知属性给出了一个匹配规则的列表,以便更进一步限定搜索服务的问题。

matchingRuleList ATTRIBUTE ::= {

WITH SYNTAX	OBJECT IDENTIFIER
EQUALITY MATCHING RULE	objectIdentifierMatch
ID	id-not-matchingRuleList }

5.12.6 过滤项 filter item

过滤项通知属性给出了在某个搜索过滤器中无效的过滤项列表。

filterItem ATTRIBUTE ::= {

WITH SYNTAX	FilterItem
ID	id-not-filterItem }

5.12.7 属性组合 attribute combinations

属性组合通知属性给出了一个在过滤器中要求出现,但实际上并未提供的属性组合的列表。

attributeCombinations ATTRIBUTE ::= {

WITH SYNTAX	AttributeCombination
ID	id-not-attributeCombinations }

5.12.8 上下文类型列表 context type list

上下文类型列表属性类型给出了一个上下文类型的列表,以便更进一步限定搜索服务的问题。

contextTypeList ATTRIBUTE ::= {

WITH SYNTAX	OBJECT IDENTIFIER
EQUALITY MATCHING RULE	objectIdentifierMatch
ID	id-not-contextTypeList }

5.12.9 上下文列表 context list

上下文列表通知属性给出了一个上下文的列表,以便更进一步限定搜索服务的问题。

```
contextList ATTRIBUTE ::= {
    WITH SYNTAX          ContextAssertion
    ID                   id-not-contextList }
```

该属性类型的值表示在某些引起该属性产生的情况下,不允许出现的一个上下文类型和该类型的某些上下文值。

5.12.10 上下文组合 context combinations

上下文组合通知属性给出了一个在过滤器中要求出现,但实际上并未提供的上下文组合的列表。

```
contextCombinations ATTRIBUTE ::= {
    WITH SYNTAX          ContextCombination
    ID                   id-not-contextCombinations }
```

5.12.11 层次选择列表 hierarchy select list

层次选择列表通知属性给出了一个标识一个或多个层次选择选项的比特串,正如在 GB/T 16264.3—2008 的 10.2.1 所定义的 HierarchySelections 结构中定义的那样。

```
hierarchySelectList ATTRIBUTE ::= {
    WITH SYNTAX          HierarchySelections
    SINGLE VALUE         TRUE
    ID                   id-not-hierarchySelectList }
```

如果在 HierarchySelection 比特串中有一个比特被设置,则表示相应的层次选择是无效的。或者是请求了一个被禁止的或不支持的选择,或者是要求的选择没有出现在请求中。

5.12.12 搜索控制选项列表 search control options list

搜索控制选项列表通知属性给出了一个比特串,标识了一个或多个搜索控制选项,这些控制选项由 GB/T 16264.3—2008 的 10.2.1 的 ASN.1 数据类型 SearchControlOptions 来定义。

```
searchControlOptionsList ATTRIBUTE ::= {
    WITH SYNTAX          SearchControlOptions
    SINGLE VALUE         TRUE
    ID                   id-not-searchControlOptionsList }
```

如果在 SearchControlOptions 比特串中有一个比特被设置,则表示相应的搜索控制选项的选择是无效的。或者是请求了一个被禁止的或不支持的选项,或者是要求的选项没有出现在请求中。

5.12.13 服务控制选项列表 service control options list

服务控制选项列表通知属性给出了一个比特串,标识了一个或多个服务控制选项,这些服务控制选项由 GB/T 16264.3—2008 的 7.5 的 ASN.1 数据类型 ServiceControlOptions 来定义。

```
serviceControlOptionsList ATTRIBUTE ::= {
    WITH SYNTAX          ServiceControlOptions
    SINGLE VALUE         TRUE
    ID                   id-not-serviceControlOptionsList }
```

如果在 ServiceControlOptions 比特串中有一个比特被设置,则它表示相应的服务控制选项的选择是无效的。或者是请求了一个被禁止的或不支持的选项,或者是要求的选项没有出现在请求中。

5.12.14 多个匹配的地理位置 multiple matching localities

多个匹配的地理位置通知属性在每个值中都标识了一个属性断言集,如果在地名词典中应用该断言将得出一个唯一的匹配。

```
multipleMatchingLocalities ATTRIBUTE ::= {
    WITH SYNTAX          MultipleMatchingLocalities
    ID                   id-not-multipleMatchingLocalities }
```

```
MultipleMatchingLocalities ::= SEQUENCE {
    matchingRuleUsed           MATCHING-RULE. &id OPTIONAL,
    attributeList               SEQUENCE OF AttributeValueAssertion }
```

元素 matchingRuleUsed 是可选的, 可用来指示所使用的基于映射的匹配规则。

没有为该属性定义匹配规则; 可允许多个相同的或近似相同的值。

5.12.15 建议张弛 proposed relaxation

建议张弛通知属性给出了元素MRMapping 的顺序, 可以作为后续 search 请求中 relaxation 组件中提供的RelaxationPolicy 的一部分。

```
proposedRelaxation ATTRIBUTE ::= {
```

WITH SYNTAX	MRMappings
ID	id-not-proposedRelaxation }

MRMappings ::= SEQUENCE OF MRMapping

MRMapping 的顺序是无关紧要的。

5.12.16 应用张弛 applied relaxation

应用张弛通知属性用来列出被放宽或被收紧的过滤器属性, 不同于那些被某个放宽策略的basic 元素所指定的属性。

```
appliedRelaxation ATTRIBUTE ::= {
```

WITH SYNTAX	OBJECT IDENTIFIER
EQUALITY MATCHING RULE	objectIdentifierMatch
ID	id-not-appliedRelaxation }

第三篇 匹配规则

6 串准备

在准备串匹配规则的评估时, 下述六步骤过程必须应用于每个当前属性值。

- 代码转换;
- 映射;
- 规范化;
- 禁止;
- 双向检查;
- 删除无意义的字符。

在任何一个步骤的失败都会导致断言结果为未定义(UNDEFINED)。

在串准备过程中创建的比较值是短暂的, 不得影响存储在目录中的属性值。

6.1 代码转换

每个非 Unicode 串值都要经过代码转换到 Unicode。

TeletexString 值被代码转换到 Unicode, 如附录 B 描述。

PrintableString 值被直接代码转换到 Unicode。

UniversalString、UTF8String 和 BMPString 的值不必进行代码转换, 因为它们是基于 Unicode 的串的(在BMPString 的情况下, 限制为 Unicode 的一个子集)。

如果在实现时不能或不愿意执行上面所描述的代码转换或代码转换失败, 则本步骤失败, 断言的赋值结果为 UNDEFINED。

被代码转换后的串为输出串。

6.2 映射

SOFT HYPHEN(U+00AD)和 MONGOLIAN TODO SOFT HYPHEN(U+1806)代码点被映

射为无。

COMBINING GRAPHEME JOINER(U+034F)和VARIATION SELECTORs(U+180B—180D,FF00—FE0F)代码点也被映射为无。OBJECT REPLACEMENT CHARACTER(U+FFFC)被映射为无。

CHARACTER TABULATION(U+0009)、LINE FEED(LF)(U+000A)、LINE TABULATION(U+000B)、FORM FEED(FF)(U+000C)、CARRIAGE RETURN(CR)(U+000D)和NEXT LINE(NEL)(U+0085)被映射为空格(U+0020)。

所有其他控制代码点(例如Cc)或具有一个控制功能的代码点(例如Cf)被映射为无。

ZERO WIDTH SPACE(U+200B)被映射为无。所有其他具有分隔符(如空格、行或段落)特性的代码点(例如Zs,Zl或Zp)被映射为空格(U+0020)。

对于不区分大小写、数字以及存储前缀串匹配规则,字符将根据RFC 3454的B.2中定义的方式进行大小写混用。

6.3 规范化

输入串被规范化为Unicode范式KC(构造时具有兼容性),正如在Unicode标准的附录#15中描述的那样。

6.4 禁止

所有未赋值的、专用的以及非字符的代码点都是被禁止的。替代代码(U+D800—DFFFF)被禁止。

REPLACEMENT CHARACTER(U+FFFD)代码被禁止。串的第一个代码点不允许是一个组合字符。空串被禁止。如果输入串包含了任意一个被禁止的代码点,则本步骤失败,且断言被赋值为UNDEFINED。输出串即为输入串。

6.5 双向检查

目前没有双向限制。输出串即为输入串。

6.6 删除无意义的字符

本步骤中,对匹配规则无意义的字符将被删除。要删除的字符根据匹配规则的不同而不同。6.6.1应用于不区分大小写和大小写精确的串匹配。6.6.2应用于numericString匹配。6.6.3应用于telephoneNumber匹配。

6.6.1 无意义空格的删除

出于本条的目的,空格被定义为SPACE(U+0020)代码点,且后面不跟组合标志。

注:前面的步骤确保了串中不包含任何分隔类的代码点,除了SPACE(U+0020)。

下面的空格被认为是无意义的,必须被删除:

——引导空格(即第一个非空格字符前的空格);

——结尾空格(即最后一个非空格字符后的空格);

——多个连续的空格(这些空格被认为与一个单独的空格字符等价,一个全部由空格组成的串与仅包含一个空格的串是等价的)。例如,从KC范式串:"〈SPACE〉〈SPACE〉foo〈SPACE〉〈SPACE〉bar〈SPACE〉〈SPACE〉"中删除空格,将导致输出串为:"foo〈SPACE〉bar",从KC范式串:"〈SPACE〉〈SPACE〉〈SPACE〉"中删除空格,将导致输出串为:"〈SPACE〉"。

6.6.2 数字串中无意义字符的删除

出于本条的目的,空格被定义为SPACE(U+0020)代码点,且后面不跟组合标志。所有的空格都被认为是无意义的,都将被删除。例如,从KC范式串:"〈SPACE〉〈SPACE〉123〈SPACE〉〈SPACE〉456〈SPACE〉〈SPACE〉"中删除空格,将导致输出串为:"123456",从KC范式串:"〈SPACE〉〈SPACE〉〈SPACE〉"中删除空格,将导致一个空的输出串。

6.6.3 电话号码中无意义字符的删除

出于本条的目的,连字符被定义为HYPHEN-MINUS(U+002D),ARMENIAN HYPHEN(U+

058A), HYPHEN(U+2010), NON-BREAKING HYPHEN(U+2011), MINUS SIGN(U+2212), SMALL HYPHENMINUS(U+FE63), 或者 FULLWIDTH HYPHEN-MINUS(U+FF0D)代码点, 且后面不跟组合标志; 空格被定义为 SPACE(U+0020)代码点, 且后面不跟组合标志。所有的连字符和空格都被认为是无意义的, 都将被删除。

7 匹配规则的定义

注: 对于 objectIdentifierMatch 和 distinguishedNameMatch 的定义, 见 GB/T 16264.2—2008。

7.1 串匹配规则

在 7.1.1 到 7.1.9 规定的匹配规则中, 所有当前的串值和存储的串值都须按照第 6 章描述的那样进行匹配准备。串准备过程将产生一个适合于逐个字符进行匹配的串。

7.1.1 大小写精确匹配和不区分大小写的匹配

大小写精确匹配规则对某个当前串和一个类型为 DirectoryString 的属性值进行相等比较, 或者和一个出现在选择类型 DirectoryString 中的数据类型之一进行相等比较, 如不考虑无意义空格的 UTF8String (见 6.6)。

所有在 Unicode 中被认为是白空格的字符必须被认为是等同于无意义的字符, 且须被忽略, 如同 6.1 开始的串匹配规则中所指定的那样。在考虑了白空格后, 应当执行不考虑大小写的匹配, 这是通过执行在 Unicode 标准中描述的大小写折叠, 并且应用在 Unicode 技术报告 15 中描述的规范化的 D 范式或 KC 范式来实现的, D 范式或 KC 范式的选择依赖于普遍检查的字符指令表和性能需求。

caseExactMatch MATCHING-RULE ::= {

SYNTAX	DirectoryString { ub-match }
ID	id-mr-caseExactMatch

不区分大小写的匹配规则对某个当前串和一个类型为 DirectoryString 的属性值进行相等比较, 或者和一个出现在选择类型 DirectoryString 中的数据类型之一(如 UTF8String)进行相等比较, 不考虑串的大小写(如“Dundee”和“DUNDEE”被认为是匹配的)以及无意义的空格(见 6.6)。除了在 6.2 讨论的串准备过程中将大写字符折叠为小写字符外, 该规则与 caseExactMatch 规则相同。

caseIgnoreMatch MATCHING-RULE ::= {

SYNTAX	DirectoryString { ub-match }
ID	id-mr-caseIgnoreMatch

如果准备好的串具有相同的长度, 且串中相应的字符都相同, 则这两个规则都返回 TRUE。

7.1.2 大小写精确排序匹配和不区分大小写的排序匹配

大小写精确排序匹配规则对某个当前串和一个类型为 DirectoryString 的属性值进行排序比较, 或者和一个出现在选择类型 DirectoryString 中的数据类型之一进行排序比较, 如不考虑无意义空格的 UTF8String (见 6.6)。

caseExactOrderingMatch MATCHING-RULE ::= {

SYNTAX	DirectoryString { ub-match }
ID	id-mr-caseExactOrderingMatch

不区分大小写的排序匹配规则对某个当前串和一个类型为 DirectoryString 的属性值进行排序比较, 或者和一个出现在选择类型 DirectoryString 中的数据类型之一(如 UTF8String)进行排序比较, 不考虑串的大小写以及无意义的空格(见 6.6)。除了在 6.2 讨论的串准备过程中将大写字符折叠为小写字符外, 该规则与 caseExactOrderingMatch 规则相同。

caseIgnoreOrderingMatch MATCHING-RULE ::= {

SYNTAX	DirectoryString { ub-match }
ID	id-mr-caseIgnoreOrderingMatch

在使用 Unicode 代码点校对排序对两个串进行比较时,如果属性值“小于”当前值或比当前值出现得早,则两个规则都返回 TRUE。

注:校对排序对某种给定语言的字符如何进行排序,提供了特定于语言和文化的信息。一个目录系统能够支持多个可配置的校对排序。该能力的实现不在本目录规范的定义范围之内。

7.1.3 大小写精确子串匹配和不区分大小写的子串匹配

大小写精确子串匹配规则判断某个当前值是否是一个类型为DirectoryString 的属性值的子串,或者是一个出现在选择类型 DirectoryString 中数据类型之一的子串,如:不考虑无意义空格的 UTF8String(见 6.6)。

caseExactSubstringsMatch MATCHING-RULE ::= {

SYNTAX	SubstringAssertion -- 仅有PrintableString 选择
ID	id-mr-caseExactSubstringsMatch }

不区分大小写的子串匹配规则判断某个当前值是否是类型为DirectoryString 的属性值的子串,或者是出现在选择类型DirectoryString 中的数据类型之一的子串,如UTF8String,不考虑串的大小写以及无意义的空格(见 6.6)。除了在 6.2 中讨论的串准备过程中将大写字符折叠为小写字符外,该规则与caseExactSubstringsMatch 规则相同。

caseIgnoreSubstringsMatch MATCHING-RULE ::= {

SYNTAX	SubstringAssertion
ID	id-mr-caseIgnoreSubstringsMatch }

SubstringAssertion ::= SEQUENCE OF CHOICE {

Initial	[0] DirectoryString {ub-match},
Any	[1] DirectoryString {ub-match},
final	[2] DirectoryString {ub-match},
control Attribute }	<ul style="list-style-type: none"> - 用于指定对下述项目的解释 - 最多有一个initial 和一个final 组件

如果属性值按照如下情况被分割为几个部分,则两个规则都返回 TRUE:

- 指定的子串(initial、any、final)按照strings 序列的次序分别与属性值的不同部分相匹配;
- 如果出现initial,则与属性值的第一部分相匹配;
- 如果出现final,则与属性值的最后一部分相匹配;
- 如果出现any,则与属性值的某个任意部分相匹配;
- 如果在子串匹配中,仅有initial、any 或final 元素被用于匹配算法中,则control 不能被用于caseIgnoreSubstringsMatch、telephoneNumberSubstringsMatch 或其他任何类型的子串匹配中;
- 如果碰到了一个control 元素,则该元素被忽略。只有在匹配规则中明确规定了控制元素在匹配算法中的用法时,才可以在那些匹配规则中使用控制元素。这样的匹配规则可能也会重新定义initial、any 和final 子串的语义。

注:匹配规则generalWordMatch 是这种匹配规则的一个示例。

在SubstringAssertion 中,最多有一个initial 和一个final。如果出现initial,则它须是第一个元素。如果出现final,则它须是最后一个元素。串中可以有零个或多个any。

要使子串中的某个组件与属性值的某一个部分相匹配,则相应的字符必须相同(包括在组合字符串序列中的所组合字符)。

7.1.4 数字串匹配

数字串匹配规则对某个当前数字串与一个类型为NumericString 的属性值进行相等比较。

numericStringMatch MATCHING-RULE ::= {

SYNTAX	NumericString
--------	---------------

ID id-mr-numericStringMatch }

除了根据 6.6.2 讨论的准备过程中将所有的空格字符都删除外,本规则与 caseIgnoreMatch 规则相同(对于数字字符而言大小写无意义)。

7.1.5 数字串排序匹配

数字串排序匹配规则对某个当前串的排列次序与类型为 NumericString 的属性值的排序次序进行比较。

numericStringOrderingMatch MATCHING-RULE ::= {

SYNTAX NumericString

ID id-mr-numericStringOrderingMatch }

除了根据 6.6.2 讨论的准备过程中将所有的空格字符都删除外,本规则与 caseIgnoreOrderingMatch 规则相同(对于数字字符而言大小写无意义)。

7.1.6 数字串子串匹配

数字串子串匹配规则判断某个当前值是否是一个类型为 NumericString 的属性值的一个子串。

numericStringSubstringsMatch MATCHING-RULE ::= {

SYNTAX SubstringAssertion

ID id-mr-numericStringSubstringsMatch }

除了根据 6.6.2 讨论的串准备过程中将所有的空格字符都删除外,本规则与 caseIgnoreSubstringsMatch 规则相同(对于数字字符而言大小写无意义)。

7.1.7 不区分大小写的列表匹配

不区分大小写的列表匹配规则对当前的某个串序列与一个类型为 DirectoryString 序列的属性值进行相等比较,而不考虑串的大小写以及无意义的空格(见 6.6)。

caseIgnoreListMatch MATCHING-RULE ::= {

SYNTAX CaseIgnoreList

ID id-mr-caseIgnoreListMatch }

CaseIgnoreList ::= SEQUENCE OF DirectoryString { ub-match }

当且仅当每个序列中的串个数相同,且相应的串都匹配时,该规则才返回 TRUE。以后的串匹配符合 caseIgnoreMatch 匹配规则。

7.1.8 不区分大小写的列表子串匹配

不区分大小写的列表子串匹配规则对当前的某个子串与类型为 DirectoryString 序列的属性值进行比较,而不考虑串的大小写和无意义的空格(见 6.6)。

caseIgnoreListSubstringsMatch MATCHING-RULE ::= {

SYNTAX SubstringAssertion

ID id-mr-caseIgnoreListSubstringsMatch }

当且仅当当前值与已存储值拼接起来形成的串相匹配时,才认为当前值与已存储值相匹配。该匹配根据 caseIgnoreSubstringsMatch 规则来完成;然而,并不认为当前值中的 initial, any 或 final 值与拼接串的某个子串相匹配,该拼接串伸展了多个已存储值的串。

7.1.9 存储前缀匹配

存储前缀匹配规则判断某个语法为 DirectoryString 的属性值是否是当前值的前缀(即初始子串),而不考虑串的大小写和无意义的空格(见 6.6)。

注:例如,它可用于在目录中,对一个电话区号与某个声称的电话号码值进行比较。

storedPrefixMatch MATCHING-RULE ::= {

SYNTAX DirectoryString { ub-match }

ID id-mr-storedPrefixMatch }

如果不考虑大小写,属性值与当前值的某个初始子串的相应字符都相同,则该规则返回 TRUE。

7.2 基于语法的匹配规则

7.2.1 布尔匹配

布尔匹配规则对当前的某个布尔值与类型为BOOLEAN 的属性值进行相等比较。

```
booleanMatch MATCHING-RULE ::= {
```

SYNTAX	BOOLEAN
ID	id-mr-booleanMatch }

如果两个值相等,即两个都是TRUE 或两个都是FALSE,则本规则返回 TRUE。

7.2.2 整数匹配

整数匹配规则对当前的某个整数值或枚举值分别与类型为INTEGER 或ENUMERATED 的属性值进行相等比较。

```
integerMatch MATCHING-RULE ::= {
```

SYNTAX	INTEGER
ID	id-mr-integerMatch }

如果当前整数值或当前枚举值与属性值相等,则本规则返回 TRUE。

7.2.3 整数排序匹配

整数排序匹配规则对当前的某个整数值与类型为INTEGER 的属性值进行比较。

```
integerOrderingMatch MATCHING-RULE ::= {
```

SYNTAX	INTEGER
ID	id-mr-integerOrderingMatch }

如果属性值小于当前值,则本规则返回 TRUE。

7.2.4 比特串匹配

比特串匹配规则对当前的某个比特串与类型为BIT STRING 的属性值进行比较。

```
bitStringMatch MATCHING-RULE ::= {
```

SYNTAX	BIT STRING
ID	id-mr-bitStringMatch }

如果属性值与当前值的比特数量相同,且各比特按位比较都匹配,则本规则返回 TRUE。如果属性语法定义为包含一个 NamedBitList,则在属性值和当前值后拖尾的零比特将被忽略。

7.2.5 八位位组串匹配

八位位组串匹配规则对当前的某个八位位组串与类型为OCTET STRING 的属性值进行相等比较。

```
octetStringMatch MATCHING-RULE ::= {
```

SYNTAX	OCTET STRING
ID	id-mr-octetStringMatch }

当且仅当八位位组串的长度相同,且相应的八位位组都相同时,本规则才返回 TRUE。

7.2.6 八位位组串排序匹配

八位位组串排序匹配规则对当前的某个八位位组串与类型为OCTET STRING 的属性值进行排序比较。

```
octetStringOrderingMatch MATCHING-RULE ::= {
```

SYNTAX	OCTET STRING
ID	id-mr-octetStringOrderingMatch }

本规则对两个八位位组串进行比较,比较顺序为从第一个八位位组到最后一个八位位组,且在八位位组内,比较顺序为从最高位比特到最低位比特。第一个出现的不同比特决定了两个串的排序。0 比

特优先于 1 比特。如果两个串相同但包含的八位位组个数不同，则较短的串优先于较长的串。

7.2.7 八位位组串子串匹配

八位位组串子串匹配规则判断当前的某个八位位组是否是类型为 OCTET STRING 的属性值的一个子串。

```
octetStringSubstringsMatch MATCHING-RULE ::= {
    SYNTAX      OctetSubstringAssertion
    ID          id-mr-octetStringSubstringsMatch h }
OctetSubstringAssertion ::= SEQUENCE OF CHOICE {
    initial     [0] OCTET STRING,
    any         [1] OCTET STRING,
    final       [2] OCTET STRING }
-- 最多一个initial 和一个final 组件
```

当属性值中包含了当前八位位组串中的八位位组序列时，则本规则返回 TRUE，如同在 caseIgnoreSubstringsMatch 中描述的那样。

7.2.8 电话号码匹配

电话号码匹配规则对某个当前值与类型为 TelephoneNumber 的属性值(见 5.7.1)进行相等比较。

```
telephoneNumberMatch MATCHING-RULE ::= {
    SYNTAX      TelephoneNumber
    ID          id-mr-telephoneNumberMatch }
```

除了在删除无意义字符的步骤中，所有的连字符和空格都无意义(见 6.6.3)并被删除外，该匹配规则与 caseIgnoreMatch 相同。

7.2.9 电话号码子串匹配

电话号码子串匹配规则判断当前的某个子串是否是类型为 PrintableString 的属性值的一个子串，该属性值为一个电话号码。

```
telephoneNumberSubstringsMatch MATCHING-RULE ::= {
    SYNTAX      SubstringAssertion
    ID          id-mr-telephoneNumberSubstringsMatch }
```

除了在删除无意义字符的步骤中，所有的连字符和空格都无意义(见 6.6.3)并被删除外，该匹配规则与 caseExactSubstringsMatch 相同。

7.2.10 表示地址匹配

表示地址匹配规则对当前的某个表示地址与类型为 PresentationAddress 的属性值进行相等比较。

```
presentationAddressMatch MATCHING-RULE ::= {
    SYNTAX      PresentationAddress
    ID          id-mr-presentationAddressMatch }
```

当且仅当当前表示地址和存储的表示地址的选择符都相同，且当前地址中的 nAddresses 是存储地址中的一个子集时，本规则返回 TRUE。

7.2.11 唯一成员匹配

唯一成员匹配规则对当前的某个唯一成员值与类型为 NameAndOptionalUID 的属性值进行相等比较。

```
uniqueMemberMatch MATCHING-RULE ::= {
    SYNTAX      NameAndOptionalUID
    ID          id-mr-uniqueMemberMatch }
```

当且仅当根据 distinguishedNameMatch 规则，属性值与当前值的 dn 组件匹配，且 uid 组件在属性值

中不存在,或者根据bitStringMatch 规则,属性值中的uid 组件与当前值中相应的组件匹配时,本规则返回 TRUE。

7.2.12 协议信息匹配

协议信息匹配规则对类型为ProtocolInformation 的当前值与相同类型的值进行相等匹配。

```
protocolInformationMatch MATCHING-RULE ::= {
```

SYNTAX	OCTET STRING
ID	id-mr-protocolInformationMatch }

断言语法的值从使用nAddress 组件的属性语法的值派生而来。

如果根据octetStringMatch 规则,当前值与已存储值的nAddress 组件相匹配,则该值返回 TRUE。

7.2.13 传真号码匹配

传真号码匹配规则对某个当前值与属性值序列中的第一个元素进行相等比较。该第一个元素为telephoneNumber,其类型为PhoneNumber (见 5.7.1)。传真号码序列中的parameters 元素不被评估。

```
facsimileNumberMatch MATCHING-RULE ::= {
```

SYNTAX	PhoneNumber
ID	id-mr-facsimileNumberMatch }

该匹配规则与telephoneNumberMatch 相同。

7.2.14 传真号码子串匹配

传真号码子串匹配规则判断某个当前子串是否是属性值序列中的第一个元素的一个子串。该第一个元素为telephoneNumber,其类型为PhoneNumber,是一个电话号码。传真号码序列中的parameters 元素不被评估。

```
facsimileNumberSubstringsMatch MATCHING-RULE ::= {
```

SYNTAX	SubstringAssertion
ID	id-mr-facsimileNumberSubstringsMatch }

该匹配规则与telephoneNumberMatch 相同。

7.2.15 UUID 对匹配

UUID 对匹配规则对类型为UUIDPair 的当前值进行相等比较,定义如下:

```
UUIDPairMatch MATCHING-RULE ::= {
```

SYNTAX	UUIDPair
ID	id-mr-uuidpairmatch }

某个类型为UUIDPair 的当前值与某个类型为UUIDPair 的目标值相匹配,当且仅当第一个值中的每个组件都与第二个值中的相应组件相等,相应组件具有相同的长度,且相应的八位位组都相等。

7.2.16 组件匹配

目录系统内属性的语法包括简单的数据类型,如文本串、整数或布尔型,也包括复杂的结构化数据类型,如目录模式操作属性的语法。通常,为复杂语法定义的匹配规则仅提供最直接有用的能力。RFC 3687 规定了通用匹配规则,该通用匹配规则可以对用户从具有任意复杂的属性语法的属性值中任意选择的组件部分进行匹配。RFC 3727 规定了一个 ASN.1 模块,可被其他规范所引用。该匹配规则规范被输入到本目录规范的SelectedAttributeTypes 中,且可能会通过 GB/T 16264.3—2008 中指定的FilterIte 的组件extensibleMatch,被选择使用。

7.3 时间匹配规则

7.3.1 UTC 时间匹配

UTC 时间匹配规则对某个当前值与一个类型为UTCTime 的属性值进行相等比较。

```
UTCTimeMatch MATCHING-RULE ::= {
```

SYNTAX	UTCTime
ID	id-mr-uTCTimeMatch }

如果属性值与当前值表示了相同的时间,则本规则返回 TRUE。如果 UTC 时间中的秒不存在,则假设秒数为零。

7.3.2 UTC 时间排序匹配

UTC 时间排序匹配规则对某个当前值与一个类型为UTCTime 的属性值进行时间排序比较。

UTCTimeOrderingMatch MATCHING-RULE ::= {

SYNTAX	UTCTime
ID	id-mr-uTCTimeOrderingMatch }

如果属性值表示的时间比当前值表示的时间早,则本规则返回 TRUE。如果 UTC 时间具有年份值为 50 到 99,则表示的时间早于具有年份值为 00 到 49 的 UTC 时间。如果 UTC 时间中的秒不存在,则假设秒数为零。

两个阿拉伯数字的年份字段值必须被合理化为 4 个阿拉伯数字的年份值,如下所示:

- 如果两个阿拉伯数字值为 00 到 49(含),则值须加 2000;以及
- 如果两个阿拉伯数字值为 50 到 99(含),则值须加 1900。

7.3.3 通用时间匹配

通用时间匹配规则对某个当前值与一个类型为GeneralizedTime (根据 GB/T 16262. 1—2006 中 42.3 的 b)或 c))的属性值进行相等比较。

generalizedTimeMatch MATCHING-RULE ::= {

SYNTAX	GeneralizedTime
-- 按照根据 GB/T 16262. 1—2006 中 42.3 的 b)或 c)	
ID	id-mr-generalizedTimeMatch }

如果属性值表示的时间与当前值表示的相同,则本规则返回 TRUE。如果所指定时间的分数或秒数不存在,则假定分数或秒数为零。

7.3.4 通用时间排序匹配

通用时间排序匹配规则对某个当前值与一个类型为GeneralizedTime (根据根据 GB/T 16262. 1—2006 中 42.3 的 b)或 c))的属性值进行时间排序比较。

generalizedTimeOrderingMatch MATCHING-RULE ::= {

SYNTAX	GeneralizedTime
-- 按照根据 GB/T 16262. 1—2006 中 42.3 的 b)或 c)	
ID	id-mr-generalizedTimeOrderingMatch }

如果属性值表示的时间比当前值表示的时间早,则本规则返回 TRUE。如果所指定时间的分数或秒数不存在,则假定分数或秒数为零。

7.3.5 系统提议的匹配

系统提议的匹配规则是一个虚拟的匹配规则,定义如下:

systemProposedMatch MATCHING-RULE ::= {

ID	id-mr-systemProposedMatch }
----	-----------------------------

请求者可将此匹配规则包含在search 请求的放宽策略中,以指示目录应当判断在匹配规则替代中应当使用哪个匹配规则。

7.4 第一组件匹配规则

7.4.1 整数第一组件匹配

整数第一组件匹配规则对当前的某个整数值与类型为SEQUENCE 的属性值进行相等比较,该属性的第一组件为必选,且类型为INTEGER。

```
integerFirstComponentMatch MATCHING-RULE ::= {
    SYNTAX INTEGER
    ID      id-mr-integerFirstComponentMatch }
```

如果属性值具有第一组件,且其值等于当前整数值,则本规则返回 TRUE。
断言语法的值从使用SEQUENCE 第一组件值的属性语法的值派生而来。

7.4.2 客体标识符第一组件匹配

客体标识符第一组件匹配规则对当前的某个客体标识符值与类型为SEQUENCE 的属性值进行相等比较,该属性的第一组件为必选,且类型为OBJECT IDENTIFIER。

```
objectIdentifierFirstComponentMatch MATCHING-RULE ::= {
    SYNTAX OBJECT IDENTIFIER
    ID      id-mr-objectIdentifierFirstComponentMatch }
```

如果属性值具有第一组件,且使用objectIdentifierMatch 规则,第一组件的值与当前的客体标识符值相匹配,则本规则返回 TRUE。

断言语法的值从使用SEQUENCE 第一组件值的属性语法的值派生而来。

7.4.3 目录串第一组件匹配

目录串第一组件匹配规则对当前的某个DirectoryString 值与类型为SEQUENCE 的属性值进行相等比较,该属性的第一组件为必选,且类型为DirectoryString。

```
directoryStringFirstComponentMatch MATCHING-RULE ::= {
    SYNTAX DirectoryString {ub-directory-string-first-component-match}
    ID      id-mr-directoryStringFirstComponentMatch }
```

如果属性值具有第一组件,且使用caseIgnoreMatch 规则,第一组件的值与当前的DirectoryString 值相匹配,则本规则返回 TRUE。

断言语法的值从使用SEQUENCE 第一组件值的属性语法的值派生而来。

7.5 字匹配规则

7.5.1 字匹配

字匹配规则对当前的某个串与类型为DirectoryString 的属性值中的字进行比较。

```
wordMatch MATCHING-RULE ::= {
    SYNTAX DirectoryString {ub-match}
    ID      id-mr-wordMatch }
```

如果当前字与属性值中的任意一个字匹配,则本规则返回 TRUE。单个字的匹配与caseIgnoreMatch 匹配规则相同。“字”的精确定义属于本地事务。

7.5.2 关键字匹配

关键字匹配规则对当前的某个串与类型为DirectoryString 的属性值中的关键字进行比较。

```
keywordMatch MATCHING-RULE ::= {
    SYNTAX DirectoryString {ub-match}
    ID      id-mr-keywordMatch }
```

如果当前值与属性值中的任意一个关键字匹配,则本规则返回 TRUE。属性值中关键字的辨别以及匹配准确性的辨别都属于本地事务。

7.5.3 通用字匹配

通用字匹配规则对当前某个串中的字与类型为DirectoryString 的属性值中的字进行比较。该匹配规则还可用于这样的属性值,即在其属性类型定义中显式地规定了DirectoryString 选择之一作为其语法。

```
generalWordMatch MATCHING-RULE ::= {
```

SYNTAX SubstringAssertion

```
ID          id-mr-generalWordMatch }
```

该匹配规则与一个普通的子串匹配规则的区别在于：在initial、any 或final 元素之前或之间插入了控制属性。如果在过滤项中没有任何控制属性，则匹配应当按照caseExactSubstringsMatch 匹配规则来执行，且initial、any 和final 元素的语义也同该匹配规则所定义的一样。然而，如果被匹配的属性类型的相等匹配规则(如果有的话)是caseIgnoreMatch，则必须使用caseIgnoreSubstringsMatch 来替代。

为通用字匹配定义了4 种类型的控制属性(对它们所处位置的限制定义如下)；任何其他的控制属性都须被忽略：

```
sequenceMatchType ATTRIBUTE ::= {
  WITH SYNTAX           SequenceMatchType
  SINGLE VALUE          TRUE
  ID                    id-cat-sequenceMatchType }-- 缺省为sequenceExact

SequenceMatchType ::= ENUMERATED {
  sequenceExact          (0),
  sequenceDeletion        (1),
  sequenceRestrictedDeletion (2),
  sequencePermutation      (3),
  sequencePermutationAndDeletion (4),
  sequenceProviderDefined (5) }

wordMatchTypes ATTRIBUTE ::= {
  WITH SYNTAX           WordMatchTypes
  SINGLE VALUE          TRUE
  ID                    id-cat-wordMatchType }-- 缺省为wordExact

WordMatchTypes ::= ENUMERATED {
  wordExact              (0),
  wordTruncated           (1),
  wordPhonetic             (2),
  wordProviderDefined      (3) }

characterMatchTypes ATTRIBUTE ::= {
  WITH SYNTAX           CharacterMatchTypes
  SINGLE VALUE          TRUE
  ID                    id-cat-characterMatchTypes }

CharacterMatchTypes ::= ENUMERATED {
  characterExact          (0),
  characterCaseIgnore       (1),
  characterMapped           (2) }

selectedContexts ATTRIBUTE ::= {
  WITH SYNTAX           ContextAssertion
  ID                    id-cat-selectedContexts }
```

每个属性都影响后续所有的initial、any 或final 元素，且它所提供的值将替代之前所适用的值。

在第一个sequenceMatchType 属性之前，如果有的话，将作为属性sequenceMatchType 的可应用的值须是sequenceExact。该属性不影响对initial 和final 元素的评估，须总是被认为与初始和终结字相匹配；它仅影响剩余的不匹配的字。initial 字，如果存在，须与存储文本的第一个字相匹配；如果两个都是

“噪音”词，则这两个字被认为是匹配的。属性sequenceMatchType 的定位(positioning)定义了匹配范式所应用的字。

注 1：出于许多实际的目的，应当能够在第一个initial 元素之前就放置sequenceMatchType；某些具体的实现可能不支持该定义的完全通用性。

在第一个wordMatchType 属性之前，如果有的话，将作为属性wordMatchType 的可应用的值须是wordExact。在第一个characterMatchType 属性之前，如果有的话，将作为属性characterMatchType 的可应用的值须是characterExact。然而，如果被匹配的属性类型所应用的相等匹配规则(如果有的话)是caseIgnoreMatch，则必须使用characterCaseIgnore 来替代。

如果控制属性selectedContexts 存在，它必须是第一个元素；且仅能有一个这样的控制属性；它须作为对已存储值的一个限制(见以下)。

如果下列条件满足，则本规则返回 TRUE：当前值包含一个非空的与指定的初始字和终结字相匹配的字序列，且在属性值中剩余的不匹配字的序列依照所指定的sequenceMatchType，即相应的字根据指定的wordMatchTypes 是相匹配的，而字内的相应字符根据指定的characterMatchTypes 是相匹配的，除了一个例外，即如果当前值中存在selectedContexts 组件，则所有的ContextAssertion 元素也都要求评估为 TRUE(正如在 GB/T 16264.2—2008 中规定的那样)。对于某个给定的已存储属性，如果字不匹配或者某些ContextAssertion 元素不匹配，则本规则返回 FALSE。

一个字是由非空格字符所组成的一个非空序列，其边界由串的起始和终止所定界，或由空格或标点符号所定界。标点符号字符被定义为不影响字所表征的语义的那些符号，一般来说，包括逗号、引号、句子结尾的句号以及括号等。判断哪些字符为标点符号应当是一个本地事务。

注 2：例如，字符“！”有时用在文本中表示一个“吸气”音(clicking sound)，如在某些非洲语言中，因此是作为字的一部分，而不是一个惊叹号(惊叹号是一个标点符号)。

类似的，final 字，如果存在，须与已存储文本的最后一个字相匹配；如果两个都是“噪音”词，则这两个字被认为是匹配的。

“噪音”词是一些这样的字，这些字与实现时定义的在语义上较弱的字列表中的一个字相匹配(例如，冠词和前置词)，根据指定的characterMatchTypes，除了要匹配initial 和final 字以外，“噪音”词在其他匹配之前被从字序列中丢弃，且如果它不是最后一个这样的规则，则wordMatchTypes 中相应的规则也被从规则序列中丢弃。

如果根据指定的sequenceMatchType，属性值中的字序列能够被转换为一个序列，该序列中包含了同当前值的字序列相同数量的字，且这些相应的字都匹配，则当前值中的字序列与属性值中的字序列相匹配。如果sequenceMatchType 取值为sequenceExact，则转换不会对序列进行改变。如果它取值为sequenceDeletion，则会从序列中删除零个或多个字。如果它取值为sequenceRestrictedDeletion，则会从序列中删除零个或多个字，但不是第一个字。如果它取值为sequencePermutation，则会改变序列中的零个或多个字的顺序。如果它取值为sequencePermutationAndDeletion，则会从序列中删除零个或多个字，并改变零个或多个剩余字的顺序。如果它取值为sequenceProviderDefined，则会根据实现时定义的规则，删除字、改变字的顺序或插入字等。

如果根据指定的wordMatchTypes 中相应的规则，属性值中的一个字能够被转换成为一个字符的序列，该序列中的字符与当前值的某个字中的字符逐次都相匹配，则当前值中的这个字与属性值中的字相匹配。每个字都应使用wordMatchTypes 中相应的规则进行匹配，且这种相应性应当在应用任何序列匹配的删除或改变顺序之前就进行判断；如果字超过wordMatchTypes 中的规则数量，则超出的字都使用最后一个规则进行匹配。如果规则是精确的，则转换不会对字进行改变。如果它取值为wordTruncated，则从字的最后一个字符开始要删除零个或多个字符，直到符合某个实现时定义的最小字长度。如果它取值为wordPhonetic，则根据实现时定义的语音匹配算法，该字被替换为与之相匹配的另一个字。如果它取值为wordProviderDefined，则根据某个实现时定义的规则进行字匹配。

每个字中的字符都使用characterMatchTypes 中相应的规则进行比较,这种相应性应当在应用任何序列匹配的删除或改变顺序之前就进行判断;如果任意一个字中的字符超过characterMatchTypes 中的规则数量,则这些超出的字符都使用最后一个规则进行匹配。如果characterMatchTypes 取值为characterExact,则若字内的相应字符相同,即认为它们是相匹配的。如果它取值为characterCaseIgnore,则若字内的相应字符在不区分大小写时相同,即认为它们是相匹配的。如果它取值为characterMapped,则若根据某个实现时定义的映射表,字内的字符都被映射到相同的字符,即认为它们是相匹配的。该映射表应当是这样的,即仅使用当前值中的字符 A~Z 和 0~9,便能够使得图 A.2/T. 51 中列出的国内字符可以匹配;而且可能会将字符的某个短序列映射为一个单独的字符,例如将 ae 映射为一个 a-e 双元音字或将 ue 映射为一个 u- 的元音变音。

7.6 近似匹配规则

7.6.1 近似串匹配

近似串匹配规则根据某个本地定义的近似匹配算法(例如拼写变量、语音匹配等),对某个当前值与一个属性值进行比较。该算法必须与处理一个类型为approximateMatch 的过滤项的响应中调用的算法相同(见 GB/T 16264.3—2008)。

approximateStringMatch MATCHING-RULE ::= {

ID id-mr-approximateStringMatch }

该匹配规则的断言语法与它所应用的属性的相等匹配规则的断言语法相同。如果没有为该属性定义相等匹配规则,则允许任何断言语法,但是该规则将总是被评估为未定义。

7.7 特殊匹配规则

7.7.1 若不存在则忽略匹配

若不存在则忽略匹配规则对任何目的及任何属性的值进行比较。

ignoreIfAbsentMatch MATCHING-RULE ::= {

ID id-mr-ignoreIfAbsentMatch }

该规则的返回如下所述:

- a) 如果属性不存在,则该规则返回值为 TRUE;
- b) 如果属性存在,则该规则返回值为未定义。

该匹配仅能够被用做是一个父匹配规则。因此,它应与当属性存在时对属性值进行匹配的匹配规则共同使用。见 GB/T 16264.2—2008 的 13.5.2。

注:在一个特定服务管理区内,通过在适当的请求属性表中指定一个空defaultValues 子组件,可获得相同的效果。

7.7.2 空匹配

空匹配规则对具有特殊规则的任何目的及任何属性的值进行比较:

nullMatch MATCHING-RULE ::= {

ID id-mr-nullMatch }

该规则的返回如下所述:

- a) 如果过滤项是非负向的,则该规则返回值为 TRUE;以及
- b) 如果过滤项是负向的,则该规则返回值为 FALSE。

该规则的正式使用可使得某个过滤项被忽略。在评估与搜索规则的兼容性时,使用空匹配的过滤项应被认为是不存在的。

7.8 区匹配

区匹配主要应用于使用了地理相关的可映射过滤项的那些search 请求。这些过滤项可以是关于localityName、stateOrProvinceName、postalCode 等的断言。

区匹配在与映射表进行匹配时使用了可组合的过滤项。

区匹配能够考虑到用户对地理位置的感知可能与 DMD 内使用的位置模型是不同的。用户的感知

与 DMD 内使用的模型之间的映射必须考虑到一个用户可能会使用那些没有直接体现在目录条目或其名(称)中的地理位置。这样的地理位置在感觉上可能是模糊的,它们没有和那些官方的地点确切地关联起来。另外,如果被查找的位置客体紧挨着临近地点的边界,则某个用户在进行搜索时可能会在猜测地理位置的名(称)时有些许的错误。出于这种目的,一个区域,例如一个国家,被划分为多个区。区是这样的一些地区,即完全地包含在某个 search 请求所提及的任何一个地理位置中。可映射过滤项的映射结果为一个区列表。关于区匹配的更详细解释,见附录 E。

当使用区匹配时,相应的映射表被成为一个地名词典(即一个地理方面的词典)。在过滤器内,一系列可组合的地理位置过滤项可能联合起来共同定义一个单独的已命名地方(即一个唯一的,通常是邻近的本地地区),或者如果允许的话,可以有少量的已命名地方都与过滤项相匹配。一个已命名地方是一个明确命名的现实世界的某个地方,例如一个城镇、村庄、国家等。

一般而言,一个地名词典将覆盖(即提供一个相关的地理数据库)组成一个国家或区域的一个域。一个地理方面的搜索请求必须根据某个具体的地名词典进行解释。搜索的范围如何判断以及选择哪个适当的地名词典属于本地事务,但是可以使用一个 DSA 缺省的地名词典来进行选择,或者基于一个或多个属性来进行选择,如与搜索操作相关联的 countryName、stateOrProvinceName 或 localityName 等(例如作为 baseObject 的辨别名中的一部分出现,或者作为过滤器的一部分出现)。

区匹配的第一步是联合使用一个或多个过滤项,来确定一个或多个已命名地方。出于这种目的,可组合的位置过滤项(即一个单独子过滤器中的所有位置过滤项)都被联合起来使用。

另外,到目前为止,此过程已经可以确定一个或多个已命名地方。在这一阶段,还根本没有使用 DIT 内的任何参考信息。此时,可使用过滤器的剩余部分在搜索范围内确定所有的条目,这些条目具有与那些已命名地方相应的地理位置,如后面所述。可能会应用放宽策略,以便已命名地方可以匹配更多的条目位置,否则可能返回的结果不够。

区匹配不支持收紧。

每个被认为是符合匹配条件的条目都必须拥有一个地点,该地点或者是由一个唯一的已命名地方所标识,可能使用多个地方名(称)值,如(“Newton”“Chester”“Cheshire”);或者是由一个或多个区(见下一段)所标识,由置于一个区属性内的值所表示。如果一个条目使用区来定义它所在的地点,则它也可能还拥有地理位置值,但是在这种情况下,该地理位置值是非正式的。管理机构有责任确保地理位置信息确实真正地标识了一个已命名地方。

区是基本的不重叠的地理组件,在性质上与地方是不同的,一个地方恰好由一个或多个区组成,如同在地名词典中所列的那样。区在地名词典的地区中是由一个串值来唯一标识的。因此,两个重叠的地方可以共享相应重叠区域的一个或多个区。区在条目中由属性表示,可能是一个操作属性。在这种情况下,区信息永远不会作为一个属性值返回,除非表示区的属性被特别请求作为一个操作属性返回。可替代的是,一个区可能是一个标准属性(例如 postalCode)。地理位置值可以如常返回,并且符合访问控制条件。

注 1: 一个区的确切特性以及与某个具体属性的映射属于本地事务,可能会依赖于具体实现的能力。在英国,作为区的较好的候选是一个邮政编码,如“RG12 2JL”,邮政编码经常定义了一个小的区域,如某个街道的一侧等。在城区,区可以很小,而在乡村,区可以相对很大。在无人居住以及无生物地区(如沙漠),一个区实际上可以非常大。

如果在为某个已命名地方所定义的区和为某个条目所定义的区之间有重叠,则认为该条目的地理位置(由区所定义)与地名词典所定义的已命名地方相匹配(即使用了一个基于重叠的匹配规则)。如果该条目的地点被定义为一个已命名地方,则该地点被认为是由组成该已命名地方的区所组成的。

区匹配允许扩展的(如放宽的)匹配,其中,级别 0 表示符合地名词典中该客体的基本定义。级别 1 以及更高的级别表示组成地方的区的逐步和系统的扩大,因此更多的条目地点会匹配。

下面是关于区匹配模型的更正式的说明:

- a) 区匹配的基础是存在一个或多个地名词典,这些地名词典由 DSA 出于某种目的而支持。一个地名词典是一个地理方面的词典,由一个适当的数据库所支持,在它范围内覆盖了一个国家或一个已命名地区。对于一个具体的搜索,对域的选择是通过本地方式来执行的。地名词典中包含了地方名(称)以及它们的特性,并列出了与之相匹配的已命名地方。这是由地点名称特性的查找和排序比较机制来支持的,地方名(称)的特性由可组合的位置属性给出,且完全独立于 DIT。
- b) 一个地名词典所覆盖的区域包含地方(*places*)。一个地方是一个公认的已命名地理地区;地方可以重叠,甚至可以扩展至区域的边界之外。通过参考地名词典而可标识的地方被称为已命名地方。
- c) 地名词典本身是基于表示地方名(称)的串。这些串用来标识(或命名)已命名地方。一个已命名地方的名(称)可以是:
 - 一个单独的地方名(称),可能由多个单词组成;
 - 多个地方名(称)的一个集合,一般来说,一个地方名(称)相应于一个较大的地区,通过该集合可以(在上下文内)将地方名(称)限定到一个相应的较小地区。
- d) 较大地区和较小地区的概念有时可能以应用到该地方的范围特性来表示。关于各种范围的地方的非正式举例有:小场地、场所、村庄、城镇、城市、县、省、国家等。一般来说,一个已命名地方须在地名词典中与容纳该地方的一个较大区域的名(称)相关联,即使不要求一个唯一的标识。
- e) 地方名(称)可能还会有一个与某个具体地方相关联的同义词,这可能会表示(如)缩略语或替代名。为每个地方定义一个规范名(称)可以带来便利,地方名(称)的同义词组件可以映射到该规范名(称)。
- f) 有时,地方名(称)可能会通过使用一些语义组件,如“靠近”,从一些更简单的地方名(称)派生而来,如“靠近 Tenterden”。可以用这种方法让人很信服地定义一个围绕英国 Kent 州 Tenterden 城周围的一个环形地方,可能最好会作为一个地方名(称)来使用,但又不是根据名(称)本身来定义一个地方。
- g) 地名词典所涵盖的所有地方都必须有一个唯一的规范名(称),可以由不同的地方名(称)集所组成,这些地方名(称)的顺序根据每个地方名(称)在上下文中所隐含的范围来进行排列。
- h) 地方被分解为区,因此区总是嵌套在每个地点内部,且地方的每个部分都有一个对应的区。一个区是地名词典中组成地方的一个组成部分;在一个地区内的每个点都有一个包含该点的单独的区。
- i) 区通常有邻区(例如,除非被某个地理特性或主要的政治特性有效阻隔,如一条湖泊、河流、大海或高山,或国家边界线等)。因此,为某个地方定义的范围一般来说可以通过将组成该地点的区的周边其他区都包含进来而进行扩展;这种扩展可以一次一步地不确定地执行。将第一级邻居包含进来的扩展称为某个地方的 1 级扩展;将更多级的扩展依次称为 2 级扩展等,依此类推。扩展的范围可以经过本地调整(扩展或缩减)以便表示一个实际的位置,但是这种调整相对来说应当是很少的。
- j) 一个表示物理客体的条目可能会被定义为拥有一个地理位置。地理位置可以在一个适当的区属性内由一系列的区来定义,或者通过使用某个位置属性,如 locationName,用一个或多个地方名(称)将其指定为一个已命名地方,属性 locationName 也可以表示为一系列的区。如果组成某个条目的地理位置的一系列区与经过查找地名词典后得出的(如上所述)表示某个地方的一系列区(可能是 n 级扩展)之间有重叠,则称此条目与该地方相匹配。
- k) 区、地方、地方名(称)的选择以及它们之间关系的编辑等属于本地事务。
- l) 如果根据条目所包含的串进行相等匹配后,条目之间是匹配的,则它们仍然是匹配的(即效果

是旁路了区匹配)。

为了更进一步限定区匹配,定义了一个非通用的信息客体类ZONAL-MATCHING,作为通用信息客体类MAPPING-BASED-MATCHING 的一个特定化。该信息客体类的一个实例将决定区匹配的特性。ZONAL-MATCHING ::= MAPPING-BASED-MATCHING{ ZonalSelect, TRUE, ZonalResult, zonalMatch, &.id }

该信息客体类的一个实例有如下特性:

- a) 虚拟引用&.selectBy,如果存在的话,本信息客体类将其替换为一系列的属性类型。本信息客体类的某个实例的选择是根据这些属性以及在搜索过滤器中表示的属性类型来决定的。一个信息客体的实例,如果它的这个组件所表示的所有属性类型都在过滤器中有表示,则该实例将被选中。不考虑属性的子类型(即选择须基于显式定义的已命名属性)。然而,在选择某个实例时,也可能会考虑一些本目录规范中未定义的其他本地标准。例如,选择可能会部分地由搜索变量中的baseObject 所决定。如果该组件不存在,则选择将完全基于本地的决策。
 - b) &.ApplicableTo 应指定一系列的与地理位置相关联的属性类型,这些属性类型由本地需求所决定,如localityName、stateOrProvinceName、streetName、postalCode 等。
 - c) 组件&.subtypeIncluded 根据本地需求而设置。
 - d) 虚拟值引用&.combinable 被无条件地设置为 TRUE。
 - e) 虚拟类型引用&.mappingResults 被本信息客体类替换为ZonalResult 数据类型。
 - f) &.userControl 根据本地需求而设置。
- 注 2: 本字段在大多数情况下须取值为 TRUE。
- g) &.exclusive 根据本地需求而设置。
- 注 3: 本信息客体类的信息客体实例将作为排斥放宽的候选。
- h) &.matching-rule 被此派生的信息客体类设置为zonalMatch ;
 - i) &.id 为区匹配算法的一个实例给定一个唯一标识符。

ZonalSelect 数据类型为:

ZonalSelect ::= SEQUENCE OF AttributeType

ZonalResult 数据类型用于指示区匹配的例外条件。

ZonalResult ::= ENUMERATED {

cannot-select-mapping	(0),
zero-mappings	(2),
multiple-mappings	(3) }

这些值分别表示:

- a) cannot-select-mapping:当基客体名(称)和子过滤器中提供的信息不足以标识在区匹配规则中所使用的映射时,则出现此例外条件。相应匹配产生的结果将是未定义。相应的,根据&.applicableTo 规范拥有可映射过滤项的子过滤器将不会被评估为TRUE。

注 4: 在一个特定服务管理区内,根据某个适当设计的搜索规则,对搜索变量的分析应该能够检测出在搜索变量中信息不够。

- b) zero-mappings:当过滤项中提供的要映射的信息不能被映射时,则出现此例外条件,不能映射的原因或者是因为在映射表中不存在相应的项,或者是因为映射过程产生了零个与条目相匹配的过滤项。在这种情况下,须返回serviceError,且所带的问题为requestedServiceNotAvailable。

CommonResults 的通知component 中须包括如下信息:

- 1) 一个通知属性searchServiceProblem,且取值为id-pr-unmatchedKeyAttributes ;以及
- 2) 一个通知属性filterItem,指示了不能提供一个匹配的可映射过滤项。

- c) multiple-mappings: 当过滤项中提供的信息能够被成功地映射到地名词典中的多个条目时, 则出现此例外条件。相应匹配所产生的结果是 TRUE, 但仍然能够引起搜索被错误中止。在这种情况下, 须返回 serviceError, 且所带的问题为 requestedServiceNotAvailable。

CommonResults 的通知 component 中须包括如下信息:

- 1) 一个通知属性 searchServiceProblem, 且取值为 id-pr-ambiguousKeyAttributes; 以及
- 2) 一个通知属性 multipleMatchingLocalities, 由匹配规则 zonalMatch 所指示。

zonalMatch 匹配规则是与信息客体类 ZONAL-MATCHING 的任意一个实例相关联的基于映射的匹配规则。

zonalMatch MATCHING-RULE ::= {

```
    UNIQUE-MATCH-INDICATOR  multipleMatchingLocalities
    ID                      id-mr-zonalMatch }
```

这个基于映射的匹配规则包括一个 UNIQUE-MATCH-INDICATOR 字段, 该字段隐含说明了针对地名词典的匹配必须得出一个无二义性的结果。如果在映射过程中有多个表条目都匹配, 则须返回 serviceError, 且所带的问题为 ambiguousKeyAttributes。通知组件 CommonResults 须包含一个 multipleMatchingLocalities 通知属性(见 5.12.14)。为每个与地名词典相匹配的表条目包含通知属性 multipleMatchingLocalities 的一个值。每个这样的值必须是一个 AttributeValueAssertion 规范的集合, 如果在每个子过滤器的逻辑与(AND)equality 过滤项中提供, 则会得到一个与相应表条目对应的唯一匹配。这就允许用户在后续的 search 请求中, 从返回的通知属性值中选择其中一个来体现在过滤器中。

第四篇 上下文

8 上下文类型的定义

本目录规范定义了若干用于目录应用的上下文类型。

8.1 语言上下文

语言上下文将某个属性值与一种具体的语言关联起来:

languageContext CONTEXT ::= {

```
    WITH SYNTAX      LanguageContextSyntax
    ID              id-avc -- language }
```

LanguageContextSyntax ::= PrintableString (SIZE(2..3)) -- 仅使用 ISO 639-2 代码

如果当前值中的字符序列与已存储值的字符序列相同, 则认为当前值与已存储值相匹配。

8.2 时间上下文

时间上下文将某个属性值与一个时间集关联起来。时间的表达可能是不同的, 包括:

- a) 绝对起始时间或终止时间(如 1994 年 12 月 14 日 24:00);
- b) 一天内的某个具体时间段(如 09:00 到 17:00);
- c) 一周内的天(如星期一);
- d) 一月内的天(如 10 号; 倒数第 2 天等);
- e) 一年内的月(如 3 月);
- f) 一个具体的年(如 1995 年);
- g) 一月内的周(如第 2 周);
- h) 周期性的日或周(如每个第 2 周);
- i) 逻辑否定(如不是星期一)。

temporalContext CONTEXT ::= {

```
    WITH SYNTAX TimeSpecification
```

ASSERTED AS TimeAssertion
ID id-avc-temporal }

TimeSpecification ::= SEQUENCE {
 time CHOICE {
 absolute SEQUENCE {
 startTime [0] GeneralizedTime OPTIONAL,
 endTime [1] GeneralizedTime OPTIONAL },
 periodic SET OF Period },
 notThisTime BOOLEAN DEFAULT FALSE,
 timeZone TimeZone OPTIONAL }
Period ::= SEQUENCE {
 timesOfDay [0] SET SIZE (1.. MAX) OF DayTimeBand OPTIONAL,
 days [1] CHOICE {
 intDay SET OF INTEGER,
 bitDay BIT STRING { sunday (0),monday (1),tuesday (2),
 wednesday (3),thursday (4),friday (5),saturday (6) },
 dayOf XDayOf } OPTIONAL,
 weeks [2] CHOICE {
 allWeeks NULL,
 intWeek SET OF INTEGER,
 bitWeek BIT STRING { week1 (0),week2 (1),week3 (2),week4 (3),
 week5 (4) } } OPTIONAL,
 months [3] CHOICE {
 allMonths NULL,
 intMonth SET OF INTEGER,
 bitMonth BIT STRING { january (0),february (1),march (2),april (3),
 may (4),june (5),july (6),august (7),september (8),
 october (9),november (10),december (11) }
 } OPTIONAL,
 years [4] SET OF INTEGER (1000 .. MAX) OPTIONAL }
XDayOf ::= CHOICE {
 first [1] NamedDay,
 second [2] NamedDay,
 third [3] NamedDay,
 fourth [4] NamedDay,
 fifth [5] NamedDay }
NamedDay ::= CHOICE {
 intNamedDays ENUMERATED {
 sunday (1),
 monday (2),
 tuesday (3),
 wednesday (4),
 Thursday (5),
 Friday (6),
 Saturday (7),
 Sunday (8)
 }
}

```

    friday      (6),
    saturday    (7) },
bitNamedDays BIT STRING { sunday (0),monday (1),tuesday (2),
                           wednesday (3),thursday (4),friday (5),saturday (6) } }

DayTimeBand ::= SEQUENCE {
    startDayTime [0] DayTime DEFAULT { hour 0 },
    endDayTime [1] DayTime DEFAULT { hour 23,minute 59,second 59 } }

DayTime ::= SEQUENCE {
    hour [0] INTEGER (0..23),
    minute [1] INTEGER (0..59) DEFAULT 0,
    second [2] INTEGER (0..59) DEFAULT 0 }

TimeZone ::= INTEGER (-12..12)

TimeAssertion ::= CHOICE {
    now NULL,
    at GeneralizedTime,
    between SEQUENCE {
        startTime [0] GeneralizedTime,
        endTime [1] GeneralizedTime OPTIONAL,
        entirely BOOLEAN DEFAULT FALSE }
}

```

time 中的 absolute 选择用绝对时间符号(GeneralizedTime)来表示了一个具体的时间或时间段。通过将 startTime 设置为与 endTime 相等,则可以表示一个具体时间。否则,如果 startTime 早于 endTime 中的时间,则表示了一个时间跨度。如果 endTime 在时间跨度中不存在,则表示包括 startTime 之后的所有时间。

periodic 允许将时间规定为时间周期的一个集合。组合效果是该集合的一个逻辑或(OR)。

注 1: 换言之,一个属性值也可以与某个具有多个上下文值的时间上下文相关联,每个上下文值定义了一个时间周期,因为这也是一个逻辑或。然而,在这里使用 SET OF 以便允许 notThisTime 可以覆盖该集合,以此可以得到一个逻辑“neither”的效果。若 notThisTime 取值为 FALSE,则选择哪种方法来规定时间周期的集合可以由规定者来选择。

在每个 Period 内,SEQUENCE OF 内的每个元素都被认为是“包含”在该 SEQUENCE OF 的后续元素之内。SEQUENCE OF 中元素的顺序是根据时间周期的粒度以升序方式来排列的,尽管并不是所有的级别都需要出现。

Period 中的最后一个元素被假定对于更高粒度的所有时间周期来说都是有效的。

注 2: 例如,如果一个 Period SEQUENCE OF timesOfDay,则它被认为对所有的日都是有效的。

timesOfDay 指示了一天内的有效的时间段,而有效的日由 Period 的下一个元素所规定。如果 days 不是下一个元素,则此时间段对下一个元素内所有可能的日都是有效的。如果没有包含 timesOfDay,则在下一个元素内一天中的所有时间都是有效的。可以通过出现多个 Period,为不同的日指定不同的时间段。

元素 days 表示了一周内、一月内或一年内的特定的日、周、月、年的选择依赖于 Period 的下一个元素。如果某个 Period 中, days 在 weeks 之前出现,则它表示一周内的几天,INTEGER 被限制取值为 1 到 7,其中 1 表示星期日。如果某个 Period 中, days 在 months 之前出现,则它表示一个月内的几天,INTEGER 被限制取值为 1 到 31,其中 1 表示一个月内的第一天。如果某个 Period 中, days 在 years 之前出现,则它表示一年中的几天,INTEGER 被限制取值为 1 到 366,其中 1 表示一年内的第一天。

dayOf 用于指示 NamedDay 在一个月内的第 1 次、第 2 次、第 3 次、第 4 次和第 5 次出现(例如该月

内的第 1 个星期一,或 8 月份的第 2 个星期二和星期五等)。使用第 5 次(fifth)总是意味着 NamedDay 在该月中的最后一次出现(例如七月的最后一个星期二)。如果在 days 中指定了 dayOf 选择,则 Period 中的 weeks 元素是无意义的,如果出现将被忽略。

如果没有指定 days,则 Period 下一个元素内的所有的日都是有效的。

元素 weeks 表示了一个月内或一年内的特定周,月、年的选择依赖于 Period 的下一个元素。如果某个 Period 中,weeks 在月之前出现,则它表示一月内的几周,INTEGER 被限制取值为 1 到 5,其中 1 表示该月内的第一周。一个月内的第一周应当是假设至少有四天包含在该月内的第一个周。第五周总是意味着该月内的最后一周。

如果某个 Period 中,weeks 在年之前出现,则它表示一年内的几周,INTEGER 被限制取值为 1 到 53,其中 1 表示该年内的第一周。一年内的第一周应当是假设至少有四天包含在该年内的第一个周。第 53 周总是意味着该年内的最后一周。

如果指定了 allWeeks,则 Period 下一个元素内的所有的周都是有效的(这也就允许使用 days 来为所有的周表示周内的天)。

如果没有指定 weeks,则 Period 下一个元素内的所有的周都是有效的。

元素 months 表示了一年内的特定月。如果 months 是由 INTEGER 表示,则 INTEGER 被限制取值为 1 到 12,其中 1 表示该年内的第一个月(即一月份)。

如果指定了 allMonths,则一年内的所有月份都有效(这也就允许使用 weeks 来为所有月表示月内的周,或者如果没有指定 weeks,则允许使用 days 来为所有的月表示月内的天)。

如果没有指定 months,则一年内的所有月都有效。

组件 years 表示了一个或多个年。如果没有指定 years,则所有的年都是有效的。

timeZone 表示了时区,以与格林威治时间(GMT)的小时差来表示,而 time 是以 GMT 来表示的。如果 timeZone 不存在,则处理此时间上下文的 DSA 将按照本 DSA 的时区来解释 time。

如果 notThisTime 取值为 FALSE,则此时间上下文值即为在 TimeSpecification 的 time 中所表示的时间。如果 notThisTime 取值为 TRUE,则此时间上下文值被认为是除了 TimeSpecification 的 time 中所表示的时间之外的所有其他时间(即执行了一个逻辑 NOT 操作)。

如果时间规范和时间断言所指定的时间之间有重叠,则认为时间断言和时间规范是相匹配的。如果时间断言中包含了 now,则当前时间被用来进行评估。如果指定了 now 或 at,且指定的时间包含在存储的 TimeSpecification 所涵盖的时间内,则此断言被评估为真。如果使用 between 和 entirely 的时间断言为 FALSE,且 between 时间段的任何部分都包含在存储的 TimeSpecification 所涵盖的时间内(这种重叠不必是完全的,如果在两个时间规范中有一部分时间周期是重叠的,则认为它们是匹配的),则此断言被赋值为真。如果使用 between 和 entirely 的时间断言为 TRUE,则仅当 between 时间段的完整部分都包含在存储的 TimeSpecification 所涵盖的时间内时,该断言才被评估为真。

举例

注 3: 当元素可以选择使用 INTEGER 或 BIT STRING 时,下述示例为元素使用了 INTEGER 格式。

a) 每天的 09:00 到 17:00,可以表示为:

```
periodic {
    timesOfDay { {
        startDayTime hour 9,
        endDayTime hour 17 } } }
```

b) 每个周一可以表示为:

```
periodic {
    days intDay : {2} }
```

c) 一月份的周一到周五的 09:00 到中午 12:00,周六全天;二月份和三月份的周二全天,可以表

示为：

```
periodic {
    timesOfDay { {
        startDayTime hour 9,
        endDayTime hour 12 } }
    days intDay : {2,3,4,5,6},
    weeks allWeeks : NULL,
    months intMonth : {1} },
    { days {7},
    weeks {1,2,3,4,5}
    months {1} }
    { days {3}
    weeks {1,2,3,4,5},
    months {2,3} }
```

d) 1996 年 8 月的全部可以表示为：

```
periodic {
    { months {8}
    years {1996} } }
```

e) 每个月的第一天可以表示为：

```
periodic {
    { days {1}
    months NULL } }
```

8.3 场所上下文

场所上下文将某个属性值与一个具体的在 POSIX 中定义的场所联系起来：

```
localeContext CONTEXT ::= {
    WITH SYNTAX LocaleContextSyntax
    ID id-avc-locale }

LocaleContextSyntax ::= CHOICE {
    localeID1OBJECT IDENTIFIER,
    localeID2DirectoryString {ub-localeContextSyntax} }
```

如果当前值与某个已存储值都是客体标识符，且这两个客体标识符相等，或者它们都是串且串相同，则认为当前值与存储值是相匹配的。

只有场所的已注册客体标识符或串才可用做上下文值。场所的概念在 ISO/IEC 9945-3:2003 中描述。

注：可以创建一个登记机构来为场所规范分配一个 OID 和/或串标识符。例如，欧洲标准化委员会，CEN，已经发布了一个关于场所信息注册的欧洲标准：ENV12005:1996，文化元素的欧洲注册过程。

8.4 LDAP 属性选项上下文

LDAP 属性选项上下文用来提供在 X.500 上下文和 LDAP 属性选项之间的融合。

```
ldapAttributeOptionContext CONTEXT ::= {
    WITH SYNTAX AttributeOptionList
    ASSERTED AS AttributeOptionList
    ABSENT-MATCH FALSE
    ID id-avc-ldapAttributeOption }
```

AttributeOptionList ::= SEQUENCE OF UTF8String

上下文值中的选项列表提供了本系列目录规范所定义的上下文概念到 ReLDAP 属性选项的最接近的、最自然的融合。每个 LDAP 的子类型化属性选项都被映射为该列表中的一个单独的UTF8String 值。如果两个ldapAttributeOptionContext 值使用不区分大小写的比较,都包含以任何顺序排列的相同的串列表,则认为这两个值是相等的。如果不区分大小写以及选项的排列顺序,某个ContextAssertion 中的AttributeOptionList 是某个已存储上下文值中的AttributeOptionList 的子集或与其相同,则认为这两个列表相匹配。

注 1: 为了简化 DER 编码,AttributeOptionList 在实现时作为一个SEQUENCE OF 来实现。

注 2: LDAP 属性选项被限制为字符“A”到“Z”、“a”到“z”、“0”到“9”以及连字符等,因此PrintableString 是足够的,而不是UTF8String。然而,属性选项的优选字符集为 UTF8,而且未来的 LDAP 扩展可能会使用更广的字符集。因此UTF8String 被选择作为经受住未来考验的规范。

特别地允许出现一个空的AttributeOptionList。在 LDAP 内,允许某个具体值同时出现在基属性内和其他任何优选的子类型中,例如(以 LDIF 格式):

```
description: This is a string  
description;lang-en: This is a string  
description;lang-en;lang-en-us: This is a string
```

在 GB/T 16264.1—2008 中,这被表示为一个单独的值“This is a string”,且伴随一个单独的上下文,其ContextType 为id-avc-ldapAttributeOption,并有三个上下文值:{ }, { "lang-en" } 和 { "lang-en", "lang-en-us" }。也就是说,一个空的AttributeOptionList、一个包含单值“lang-en”的AttributeOptionList 以及一个包含两个值“lang-en”和“lang-en-us”的AttributeOptionList。

附录 A
(规范性附录)
用 ASN.1 描述的选定的属性类型

本附录包括在本目录规范中涵盖的所有 ASN.1 类型和值的定义,形式为 ASN.1 模块 SelectedAttributeTypes。

SelectedAttributeTypes {joint-iso-itu-t ds(5) module(1) selectedAttributeTypes(5) 5}

DEFINITIONS ::=

BEGIN

-- EXPORTS All --

-- 本模块中定义的类型和值输出可用于目录规范包含的其他 ASN.1 模块,

-- 以及使用它们访问目录服务的其他应用。

-- 其他应用可把它们用于自己的目的,

-- 但这并不限制为维护或改进目录服务所需的扩充和修改。

IMPORTS

-- 来自 GB/T 16264. 2—2008

```
directoryAbstractService, id-at, id-avc, id-cat, id-mr, id-not, id-pr, informationFramework,
serviceAdministration, upperBounds
    FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5 }
Attribute, ATTRIBUTE, AttributeType, AttributeValueAssertion, CONTEXT, ContextAssertion,
DistinguishedName, distinguishedNameMatch, MAPPING-BASED-MATCHING{ },
MATCHING-RULE, OBJECT-CLASS, objectIdentifierMatch
    FROM InformationFramework informationFramework
AttributeCombination, ContextCombination, MRMapping
    FROM ServiceAdministration serviceAdministration
```

-- 来自 GB/T 16264. 3—2008

```
FilterItem, HierarchySelections, SearchControlOptions, ServiceControlOptions
    FROM DirectoryAbstractService directoryAbstractService
```

-- 来自 GB/T 16264. 6—2008

```
ub-answerback, ub-business-category, ub-common-name, ub-country-code, ub-description,
ub-destination-indicator, ub-directory-string-first-component-match, ub-international-isdn-number,
ub-knowledge-information, ub-localeContextSyntax, ub-locality-name, ub-match, ub-name,
ub-organization-name, ub-organizational-unit-name, ub-physical-office-name, ub-postal-code,
ub-postal-line, ub-postal-string, ub-post-office-box, ub-pseudonym, ub-serial-number, ub-state-name,
ub-street-address, ub-surname, ub-telephone-number, ub-telex-number, ub-teletex-terminal-id,
ub-title, ub-user-password, ub-x121-address
```

FROM UpperBounds upperBounds

-- 来自 ISO/IEC 10021-4

G3FacsimileNonBasicParameters

```
    FROM MTSAbstractService{joint-iso-itu-t mhs(6) mts(3) modules(0)
mts-abstract-service(1) version-1999(1) } ;
```

/* 出自 IETF RFC 3727

下列输入仅提供用于信息(见 7.2.16),它不由这些目录规范内的任何 ASN.1 构造引用。注意 RFC 3727 的 ASN.1 模块从 GB/T 16264.2—2008 第 4 版的 *InformationFramework* 模块输入。从这些目录规范和 RFC 3727 输入的规范应采取纠正动作,例如产生 RFC 3727 的 ASN.1 模块的副本,然后更新 IMPORT 语句。

```
allComponentsMatch, componentFilterMatch, directoryComponentsMatch, presentMatch, rdnMatch
    FROM ComponentMatching {iso(1) 2 36 79672281 xed(3) module (0)
        component-matching(4)} */
```

-- 目录串类型 --

```
DirectoryString { INTEGER : maxSize } ::= CHOICE {
    teletexString TeletexString (SIZE (1..maxSize)),
    printableString     PrintableString (SIZE (1..maxSize)),
    bmpString          BMPString (SIZE (1..maxSize)),
    universalString    UniversalString (SIZE (1..maxSize)),
    uTF8String         UTF8String (SIZE (1..maxSize)) }
```

-- 属性类型 --

```
knowledgeInformation ATTRIBUTE ::= {
    WITH SYNTAX           DirectoryString {ub-knowledge-information}
    EQUALITY MATCHING RULE
    ID                     caseIgnoreMatch
                           id-at-knowledgeInformation }

name ATTRIBUTE ::= {
    WITH SYNTAX           DirectoryString {ub-name}
    EQUALITY MATCHING RULE
    SUBSTRINGS MATCHING RULE
    ID                     caseIgnoreSubstringsMatch
                           id-at-name }

commonName ATTRIBUTE ::= {
    SUBTYPE OF             DirectoryString {ub-common-name}
    WITH SYNTAX           id-at-commonName }

surname ATTRIBUTE ::= {
    SUBTYPE OF             name
    WITH SYNTAX           DirectoryString {ub-surname}
    ID                     id-at-surname }

givenName ATTRIBUTE ::= {
    SUBTYPE OF             name
    WITH SYNTAX           DirectoryString {ub-name}
    ID                     id-at-givenName }

initials ATTRIBUTE ::= {
    SUBTYPE OF             name
    WITH SYNTAX           DirectoryString {ub-name}
    ID                     id-at-initials }

generationQualifier ATTRIBUTE ::= {
    SUBTYPE OF             name
    WITH SYNTAX           DirectoryString {ub-name} }
```

```

ID id-at-generationQualifier }

uniqueIdentifier ATTRIBUTE ::= {
    WITH SYNTAX UniqueIdentifier
    EQUALITY MATCHING RULE bitStringMatch
    ID id-at-uniqueIdentifier }

UniqueIdentifier ::= BIT STRING

dnQualifier ATTRIBUTE ::= {
    WITH SYNTAX PrintableString
    EQUALITY MATCHING RULE caseIgnoreMatch
    ORDERING MATCHING RULE caseIgnoreOrderingMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID id-at-dnQualifier }

serialNumber ATTRIBUTE ::= {
    WITH SYNTAX PrintableString (SIZE (1.. ub-serial-number))
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID id-at-serialNumber }

pseudonym ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX DirectoryString {ub-pseudonym}
    ID id-at-pseudonym }

UUIDPair ATTRIBUTE ::= {
    WITH SYNTAX UUIDPair
    EQUALITY MATCHING RULE UUIDPairMatch
    ID id-at-t-uuidpair }

UUIDPair ::= SEQUENCE {
    issuerUUID UUID,
    subjectUUID UUID }

UUID ::= OCTET STRING (SIZE(16))--只有 UUID 格式

countryName ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX CountryName
    SINGLE VALUE TRUE
    ID id-at-countryName }

CountryName ::= PrintableString (SIZE(2))-- 只有 GB/T 2659—2000 码

localityName ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX DirectoryString {ub-locality-name}
    ID id-at-localityName }

collectiveLocalityName ATTRIBUTE ::= {
    SUBTYPE OF localityName
    COLLECTIVE TRUE
    ID id-at-collectiveLocalityName }

```

```
stateOrProvinceName ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX DirectoryString { ub-state-name }
    ID id-at-stateOrProvinceName }

collectiveStateOrProvinceName ATTRIBUTE ::= {
    SUBTYPE OF stateOrProvinceName
    COLLECTIVE TRUE
    ID id-at-collectiveStateOrProvinceName }

streetAddress ATTRIBUTE ::= {
    WITH SYNTAX DirectoryString { ub-street-address }
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID id-at-streetAddress }

collectiveStreetAddress ATTRIBUTE ::= {
    SUBTYPE OF streetAddress
    COLLECTIVE TRUE
    ID id-at-collectiveStreetAddress }

houseIdentifier ATTRIBUTE ::= {
    WITH SYNTAX DirectoryString { ub-name }
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID id-at-houseIdentifier }

organizationName ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX DirectoryString { ub-organization-name }
    ID id-at-organizationName }

collectiveOrganizationName ATTRIBUTE ::= {
    SUBTYPE OF organizationName
    COLLECTIVE TRUE
    ID id-at-collectiveOrganizationName }

organizationalUnitName ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX DirectoryString { ub-organizational-unit-name }
    ID id-at-organizationalUnitName }

collectiveOrganizationalUnitName ATTRIBUTE ::= {
    SUBTYPE OF organizationalUnitName
    COLLECTIVE TRUE
    ID id-at-collectiveOrganizationalUnitName IUnitName }

title ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX DirectoryString { ub-title }
    ID id-at-title }

description ATTRIBUTE ::= {
```

```

WITH SYNTAX                               DirectoryString {ub-description}
EQUALITY MATCHING RULE                   caseIgnoreMatch
SUBSTRINGS MATCHING RULE                caseIgnoreSubstringsMatch
ID                                         id-at-description }

searchGuide ATTRIBUTE ::= {
  WITH                                         SYNTAX Guide
  ID                                           id-at-searchGuide }

Guide ::= SET {
  objectClass      [0]          OBJECT-CLASS. &id OPTIONAL,
  criteria         [1]          Criteria }

Criteria ::= CHOICE {
  type            [0]          CriteriaItem,
  and             [1]          SET OF Criteria,
  or              [2]          SET OF Criteria,
  not             [3]          Criteria }

CriteriaItem ::= CHOICE {
  equality        [0]          AttributeType,
  substrings      [1]          AttributeType,
  greaterOrEqual  [2]          AttributeType,
  lessOrEqual     [3]          AttributeType,
  approximateMatch [4]         AttributeType }

enhancedSearchGuide ATTRIBUTE ::= {
  WITH SYNTAX
  ID                                         EnhancedGuide
  id-at-enhancedSearchGuide }

EnhancedGuide ::= SEQUENCE {
  objectClass      [0]          OBJECT-CLASS. &id,
  criteria         [1]          Criteria,
  subset           [2]          INTEGER
  { baseObject (0), oneLevel (1), wholeSubtree (2) } DEFAULT oneLevel }

businessCategory ATTRIBUTE ::= {
  WITH SYNTAX                               DirectoryString {ub-business-category}
  EQUALITY MATCHING RULE                   caseIgnoreMatch
  SUBSTRINGS MATCHING RULE                caseIgnoreSubstringsMatch
  ID                                         id-at-businessCategory }

postalAddress ATTRIBUTE ::= {
  WITH SYNTAX                               PostalAddress
  EQUALITY MATCHING RULE                   caseIgnoreListMatch
  SUBSTRINGS MATCHING RULE                caseIgnoreListSubstringsMatch
  ID                                         id-at-postalAddress }

PostalAddress ::= SEQUENCE SIZE(1.. ub-postal-line) OF DirectoryString
  {ub-postal-string}

collectivePostalAddress ATTRIBUTE ::= {
  SUBTYPE OF                                postalAddress

```

```

COLLECTIVE                         TRUE
ID                                 id-at-collectivePostalAddress }

postalCode ATTRIBUTE ::= {
    WITH SYNTAX
    EQUALITY MATCHING RULE
    SUBSTRINGS MATCHING RULE
    ID
    collectivePostalCode ATTRIBUTE ::= {
        SUBTYPE OF
        COLLECTIVE
        ID
        postalCode
        TRUE
        id-at-collectivePostalCode }
    postOfficeBox ATTRIBUTE ::= {
        WITH SYNTAX
        EQUALITY MATCHING RULE
        SUBSTRINGS MATCHING RULE
        ID
        collectivePostOfficeBox ATTRIBUTE ::= {
            SUBTYPE OF
            COLLECTIVE
            ID
            postOfficeBox
            TRUE
            id-at-collectivePostOfficeBox }
    physicalDeliveryOfficeName ATTRIBUTE ::= {
        WITH SYNTAX
        EQUALITY MATCHING RULE
        SUBSTRINGS MATCHING RULE
        ID
        physicalDeliveryOfficeName
        TRUE
        id-at-physicalDeliveryOfficeName }
    collectivePhysicalDeliveryOfficeName ATTRIBUTE ::= {
        SUBTYPE OF
        COLLECTIVE
        ID
        physicalDeliveryOfficeName
        TRUE
        id-at-collectivePhysicalDeliveryOfficeName }
    telephoneNumber ATTRIBUTE ::= {
        WITH SYNTAX
        EQUALITY MATCHING RULE
        SUBSTRINGS MATCHING RULE
        ID
        TelephoneNumber
        telephoneNumberMatch
        telephoneNumberSubstringsMatch
        id-at-telephoneNumber }
    TelephoneNumber ::= PrintableString (SIZE(1.. ub-telephone-number))
    -- 只符合 ITU-T E.123 建议书的串
    collectiveTelephoneNumber ATTRIBUTE ::= {
        SUBTYPE OF
        COLLECTIVE
        ID
        telephoneNumber
        TRUE
        id-at-collectiveTelephoneNumber }
    telexNumber ATTRIBUTE ::= {
        WITH SYNTAX
        ID
        TelexNumber
        id-at-telexNumber }

```

```

TelexNumber ::= SEQUENCE {
    telexNumber                  PrintableString (SIZE (1.. ub-telex-number)),
    countryCode                  PrintableString (SIZE (1.. ub-country-code)),
    answerback                   PrintableString (SIZE (1.. ub-answerback)) }

collectiveTelexNumber ATTRIBUTE ::= {
    SUBTYPE OF              telexNumber
    COLLECTIVE               TRUE
    ID                      id-at-collectiveTelexNumber }

facsimileTelephoneNumber ATTRIBUTE ::= {
    WITH SYNTAX           FacsimileTelephoneNumber
    EQUALITY MATCHING RULE facsimileNumberMatch
    SUBSTRINGS MATCHING RULE facsimileNumberSubstringsMatch
    ID                     id-at-facsimileTelephoneNumber }

FacsimileTelephoneNumber ::= SEQUENCE {
    telephoneNumber          TelephoneNumber,
    parameters                G3FacsimileNonBasicParameters OPTIONAL }

collectiveFacsimileTelephoneNumber ATTRIBUTE ::= {
    SUBTYPE OF              facsimileTelephoneNumber
    COLLECTIVE               TRUE
    ID                      id-at-collectiveFacsimileTelephoneNumber }

x121Address ATTRIBUTE ::= {
    WITH SYNTAX           X121Address
    EQUALITY MATCHING RULE numericStringMatch
    SUBSTRINGS MATCHING RULE numericStringSubstringsMatch
    ID                     id-at-x121Address }

X121Address ::= NumericString (SIZE(1.. ub-x121-address))
-- 正如 ITU-T X.121 建议书所规定的串

internationalISDNNumber ATTRIBUTE ::= {
    WITH SYNTAX           InternationalISDNNumber
    EQUALITY MATCHING RULE numericStringMatch
    SUBSTRINGS MATCHING RULE numericStringSubstringsMatch
    ID                     id-at-internationalISDNNumber }

InternationalISDNNumber ::= NumericString (SIZE(1.. ub-international-isdn-number))
-- 只符合 ITU-T E.164 建议书的串

collectiveInternationalISDNNumber ATTRIBUTE ::= {
    SUBTYPE OF              internationalISDNNumber
    COLLECTIVE               TRUE
    ID                      id-at-collectiveInternationalISDNNumber }

registeredAddress ATTRIBUTE ::= {
    SUBTYPE OF              postalAddress
    WITH SYNTAX             PostalAddress
    ID                     id-at-registeredAddress }

destinationIndicator ATTRIBUTE ::=

```

```

WITH SYNTAX          DestinationIndicator
EQUALITY MATCHING RULE   caseIgnoreMatch
SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
ID                  id-at-destinationIndicator }

DestinationIndicator ::= PrintableString (SIZE(1.. ub-destination-indicator))
-- 只有字母字符

communicationsService ATTRIBUTE ::= {
  WITH SYNTAX          OBJECT IDENTIFIER
  EQUALITY MATCHING RULE  objectIdentifierMatch
  ID                  id-at-communicationsService }

communicationsNetwork ATTRIBUTE ::= {
  WITH SYNTAX          OBJECT IDENTIFIER
  EQUALITY MATCHING RULE  objectIdentifierMatch
  SINGLE VALUE          TRUE
  ID                  id-at-communicationsNetwork }

preferredDeliveryMethod ATTRIBUTE ::= {
  WITH SYNTAX          PreferredDeliveryMethod
  SINGLE VALUE          TRUE
  ID                  id-at-preferredDeliveryMethod }

PreferredDeliveryMethod ::= SEQUENCE OF INTEGER
any-delivery-method      (0),
mhs-delivery             (1),
physical-delivery        (2),
telex-delivery           (3),
teletex-delivery         (4),
g3-facsimile-delivery   (5),
g4-facsimile-delivery   (6),
ia5-terminal-delivery   (7),
videotex-delivery        (8),
telephone-delivery       (9) }

presentationAddress ATTRIBUTE ::= {
  WITH SYNTAX          PresentationAddress
  EQUALITY MATCHING RULE  presentationAddressMatch
  SINGLE VALUE          TRUE
  ID                  id-at-presentationAddress }

PresentationAddress ::= SEQUENCE {
  pSelector      [0] OCTET STRING OPTIONAL,
  sSelector      [1] OCTET STRING OPTIONAL,
  tSelector      [2] OCTET STRING OPTIONAL,
  nAddresses    [3] SET SIZE (1.. MAX) OF OCTET STRING }

supportedApplicationContext ATTRIBUTE ::= {
  WITH SYNTAX          OBJECT IDENTIFIER
  EQUALITY MATCHING RULE  objectIdentifierMatch
}

```

```

ID
protocolInformation ATTRIBUTE ::= {
    WITH SYNTAX
    EQUALITY MATCHING RULE
    ID
ProtocolInformation ::= SEQUENCE {
    nAddress
    profiles SET OF
distinguishedName ATTRIBUTE ::= {
    WITH SYNTAX
    EQUALITY MATCHING RULE
    ID
member ATTRIBUTE ::= {
    SUBTYPE OF
    ID
uniqueMember ATTRIBUTE ::= {
    WITH SYNTAX
    EQUALITY MATCHING RULE
    ID
NameAndOptionalUID ::= SEQUENCE {
    dn
    uid
owner ATTRIBUTE ::= {
    SUBTYPE OF
    ID
roleOccupant ATTRIBUTE ::= {
    SUBTYPE OF
    ID
seeAlso ATTRIBUTE ::= {
    SUBTYPE OF
    ID
dmdName ATTRIBUTE ::= {
    SUBTYPE OF
    WITH SYNTAX
    ID
-- 通知属性 --
dSAPrivate ATTRIBUTE ::= {
    WITH SYNTAX
    EQUALITY MATCHING RULE
    ID
searchServiceProblem ATTRIBUTE ::= {
    WITH SYNTAX
    EQUALITY MATCHING RULE
    id-at-supportedApplicationContext }

ProtocolInformation
protocolInformationMatch
id-at-protocolInformation }

OCTET STRING,
OBJECT IDENTIFIER }

DistinguishedName
distinguishedNameMatch
id-at-distinguishedName }

distinguishedName
id-at-member }

NameAndOptionalUID
uniqueMemberMatch
id-at-uniqueMember }

DistinguishedName,
UniqueIdentifier OPTIONAL }

distinguishedName
id-at-owner }

distinguishedName
id-at-roleOccupant }

distinguishedName
id-at-seeAlso }

name
DirectoryString{ub-common-name}
id-at-dmdName }

OBJECT IDENTIFIER
objectIdentifierMatch
id-not-dSAPrivate }

OBJECT IDENTIFIER
objectIdentifierMatch

```

```

SINGLE VALUE TRUE
ID id-not-searchServiceProblem }

serviceType ATTRIBUTE ::= {
  WITH SYNTAX OBJECT IDENTIFIER
  EQUALITY MATCHING RULE objectIdentifierMatch
  SINGLE VALUE TRUE
  ID id-not-serviceType }

attributeTypeList ATTRIBUTE ::= {
  WITH SYNTAX OBJECT IDENTIFIER
  EQUALITY MATCHING RULE objectIdentifierMatch
  ID id-not-attributeTypeList }

matchingRuleList ATTRIBUTE ::= {
  WITH SYNTAX OBJECT IDENTIFIER
  EQUALITY MATCHING RULE objectIdentifierMatch
  ID id-not-matchingRuleList }

filterItem ATTRIBUTE ::= {
  WITH SYNTAX FilterItem
  ID id-not-filterItem }

attributeCombinations ATTRIBUTE ::= {
  WITH SYNTAX AttributeCombination
  ID id-not-attributeCombinations }

contextTypeList ATTRIBUTE ::= {
  WITH SYNTAX OBJECT IDENTIFIER
  EQUALITY MATCHING RULE objectIdentifierMatch
  ID id-not-contextTypeList }

contextList ATTRIBUTE ::= {
  WITH SYNTAX ContextAssertion
  ID id-not-contextList }

contextCombinations ATTRIBUTE ::= {
  WITH SYNTAX ContextCombination
  ID id-not-contextCombinations }

hierarchySelectList ATTRIBUTE ::= {
  WITH SYNTAX HierarchySelections
  SINGLE VALUE TRUE
  ID id-not-hierarchySelectList }

searchControlOptionsList ATTRIBUTE ::= {
  WITH SYNTAX SearchControlOptions
  SINGLE VALUE TRUE
  ID id-not-searchControlOptionsList }

serviceControlOptionsList ATTRIBUTE ::= {
  WITH SYNTAX ServiceControlOptions
  SINGLE VALUE TRUE
  ID id-not-serviceControlOptionsList }

```

```

multipleMatchingLocalities ATTRIBUTE ::= {
    WITH SYNTAX           MultipleMatchingLocalities
    ID                     id-not-multipleMatchingLocalities }

MultipleMatchingLocalities ::= SEQUENCE {
    matchingRuleUsed      MATCHING-RULE. &id OPTIONAL,
    attributeList SEQUENCE OF AttributeValueAssertion }

proposedRelaxation ATTRIBUTE ::= {
    WITH SYNTAX SEQUENCE OF MRMapping
    ID                   id-not-proposedRelaxation }

appliedRelaxation ATTRIBUTE ::= {
    WITH SYNTAX           OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                   id-not-appliedRelaxation }

-- 匹配规则 --
caseIgnoreMatch MATCHING-RULE ::= {
    SYNTAX           DirectoryString {ub-match}
    ID               id-mr-caseIgnoreMatch }

caseIgnoreOrderingMatch MATCHING-RULE ::= {
    SYNTAX           DirectoryString {ub-match}
    ID               id-mr-caseIgnoreOrderingMatch }

caseIgnoreSubstringsMatch MATCHING-RULE ::= {
    SYNTAX           SubstringAssertion
    ID               id-mr-caseIgnoreSubstringsMatch }

SubstringAssertion ::= SEQUENCE OF CHOICE {
    initial          [0]     DirectoryString {ub-match},
    any              [1]     DirectoryString {ub-match},
    final            [2]     DirectoryString {ub-match},
    control Attribute } -- 用于规定下列项的解释
    -- 最多有一个initial 和一个final 组件

caseExactMatch MATCHING-RULE ::= {
    SYNTAX           DirectoryString {ub-match}
    ID               id-mr-caseExactMatch }

caseExactOrderingMatch MATCHING-RULE ::= {
    SYNTAX           DirectoryString {ub-match}
    ID               id-mr-caseExactOrderingMatch }

caseExactSubstringsMatch MATCHING-RULE ::= {
    SYNTAX SubstringAssertion -- 只能选择PrintableString
    ID               id-mr-caseExactSubstringsMatch }

numericStringMatch MATCHING-RULE ::= {
    SYNTAX NumericString
    ID               id-mr-numericStringMatch }

numericStringOrderingMatch MATCHING-RULE ::= {
    SYNTAX NumericString

```

```
ID id-mr-numericStringOrderingMatch }
numericStringSubstringsMatch MATCHING-RULE ::= {
    SYNTAX SubstringAssertion
    ID id-mr-numericStringSubstringsMatch }
caseIgnoreListMatch MATCHING-RULE ::= {
    SYNTAX CaseIgnoreList
    ID id-mr-caseIgnoreListMatch }
CaseIgnoreList ::= SEQUENCE OF DirectoryString {ub-match}
caseIgnoreListSubstringsMatch MATCHING-RULE ::= {
    SYNTAX SubstringAssertion
    ID id-mr-caseIgnoreListSubstringsMatch }
storedPrefixMatch MATCHING-RULE ::= {
    SYNTAX DirectoryString {ub-match}
    ID id-mr-storedPrefixMatch }
booleanMatch MATCHING-RULE ::= {
    SYNTAX BOOLEAN
    ID id-mr-booleanMatch }
integerMatch MATCHING-RULE ::= {
    SYNTAX INTEGER
    ID id-mr-integerMatch }
integerOrderingMatch MATCHING-RULE ::= {
    SYNTAX INTEGER
    ID id-mr-integerOrderingMatch }
bitStringMatch MATCHING-RULE ::= {
    SYNTAX BIT STRING
    ID id-mr-bitStringMatch }
octetStringMatch MATCHING-RULE ::= {
    SYNTAX OCTET STRING
    ID id-mr-octetStringMatch }
octetStringOrderingMatch MATCHING-RULE ::= {
    SYNTAX OCTET STRING
    ID id-mr-octetStringOrderingMatch }
octetStringSubstringsMatch MATCHING-RULE ::= {
    SYNTAX OctetSubstringAssertion
    ID id-mr-octetStringSubstringsMatch }
OctetSubstringAssertion ::= SEQUENCE OF CHOICE {
    initial [0] OCTET STRING,
    any [1] OCTET STRING,
    final [2] OCTET STRING }
-- 最多由一个initial 和一个final 构成
telephoneNumberMatch MATCHING-RULE ::= {
    SYNTAX TelephoneNumber
    ID id-mr-telephoneNumberMatch }
```

```

telephoneNumberSubstringsMatch MATCHING-RULE ::= {
    SYNTAX SubstringAssertion
    ID          id-mr-telephoneNumberSubstringsMatch }

presentationAddressMatch MATCHING-RULE ::= {
    SYNTAX PresentationAddress
    ID          id-mr-presentationAddressMatch }

uniqueMemberMatch MATCHING-RULE ::= {
    SYNTAX NameAndOptionalUID
    ID          id-mr-uniqueMemberMatch }

protocolInformationMatch MATCHING-RULE ::= {
    SYNTAX OCTET STRING
    ID          id-mr-protocolInformationMatch }

facsimileNumberMatch MATCHING-RULE ::= {
    SYNTAX TelephoneNumber
    ID          id-mr-facsimileNumberMatch }

facsimileNumberSubstringsMatch MATCHING-RULE ::= {
    SYNTAX SubstringAssertion
    ID          id-mr-facsimileNumberSubstringsMatch }

UUIDPairMatch MATCHING-RULE ::= {
    SYNTAX UUIDPair
    ID          id-mr-uuidpairmatch }

UTCTimeMatch MATCHING-RULE ::= {
    SYNTAX UTCTime
    ID          id-mr-uTCTimeMatch }

UTCTimeOrderingMatch MATCHING-RULE ::= {
    SYNTAX UTCTime
    ID          id-mr-uTCTimeOrderingMatch }

generalizedTimeMatch MATCHING-RULE ::= {
    SYNTAX GeneralizedTime
    -- 按照 GB/T 16262.1—2006 的 42.3b)或 c)
    ID          id-mr-generalizedTimeMatch }

generalizedTimeOrderingMatch MATCHING-RULE ::= {
    SYNTAX GeneralizedTime
    -- 按照 GB/T 16262.1—2006 的 42.3 b)或 c)
    ID          id-mr-generalizedTimeOrderingMatch }

systemProposedMatch MATCHING-RULE ::= {
    ID          id-mr-systemProposedMatch }

integerFirstComponentMatch MATCHING-RULE ::= {
    SYNTAX INTEGER
    ID          id-mr-integerFirstComponentMatch }

objectIdentifierFirstComponentMatch MATCHING-RULE ::= {
    SYNTAX OBJECT IDENTIFIER
    ID          id-mr-objectIdentifierFirstComponentMatch }

```

```
directoryStringFirstComponentMatch MATCHING-RULE ::= {
    SYNTAX DirectoryString { ub-directory-string-first-component-match }
    ID          id-mr-directoryStringFirstComponentMatch }

wordMatch MATCHING-RULE ::= {
    SYNTAX DirectoryString { ub-match }
    ID          id-mr-wordMatch }

keywordMatch MATCHING-RULE ::= {
    SYNTAX DirectoryString { ub-match }
    ID          id-mr-keywordMatch }

generalWordMatch MATCHING-RULE ::= {
    SYNTAX SubstringAssertion
    ID          id-mr-generalWordMatch }

sequenceMatchType ATTRIBUTE ::= {
    WITH SYNTAX SequenceMatchType
    SINGLE VALUE TRUE
    ID          id-cat-sequenceMatchType }-- 缺省为 sequenceExact

SequenceMatchType ::= ENUMERATED {
    sequenceExact          (0),
    sequenceDeletion        (1),
    sequenceRestrictedDeletion (2),
    sequencePermutation     (3),
    sequencePermutationAndDeletion (4),
    sequenceProviderDefined (5) }

wordMatchTypes ATTRIBUTE ::= {
    WITH SYNTAX WordMatchTypes
    SINGLE VALUE TRUE
    ID          id-cat-wordMatchType }-- 缺省为 wordExact

WordMatchTypes ::= ENUMERATED {
    wordExact          (0),
    wordTruncated       (1),
    wordPhonetic         (2),
    wordProviderDefined (3) }

characterMatchTypes ATTRIBUTE ::= {
    WITH SYNTAX CharacterMatchTypes
    SINGLE VALUE TRUE
    ID          id-cat-characterMatchTypes }

CharacterMatchTypes ::= ENUMERATED {
    characterExact          (0),
    characterCaseIgnore      (1),
    characterMapped          (2) }

selectedContexts ATTRIBUTE ::= {
    WITH SYNTAX ContextAssertion
    ID          id-cat-selectedContexts }
```

```

approximateStringMatch MATCHING-RULE ::= {
    ID           id-mr-approximateStringMatch imateStringMatch }

ignoreIfAbsentMatch MATCHING-RULE ::= {
    ID           id-mr-ignoreIfAbsentMatch }

nullMatch MATCHING-RULE ::= {
    ID           id-mr-nullMatch }

ZONAL-MATCHING ::= MAPPING-BASED-MATCHING { ZonalSelect, TRUE, Zonal-
Result, zonalMatch, &.id }

ZonalSelect ::= SEQUENCE OF AttributeType

ZonalResult ::= ENUMERATED {
    cannot-select-mapping          (0),
    zero-mappings                  (2),
    multiple-mappings              (3) }

zonalMatch MATCHING-RULE ::= {
    UNIQUE-MATCH-INDICATOR multipleMatchingLocalities
    ID           id-mr-zonalMatch }

-- 上下文 --
languageContext CONTEXT ::= {
    WITH SYNTAX LanguageContextSyntax
    ID           id-avc-language }

LanguageContextSyntax ::= PrintableString (SIZE(2..3)) -- 只有 ISO 639-2 码

temporalContext CONTEXT ::= {
    WITH SYNTAX TimeSpecification
    ASSERTED AS TimeAssertion
    ID           id-avc-temporal }

TimeSpecification ::= SEQUENCE {
    time   CHOICE {
        absolute SEQUENCE {
            startTime [0] GeneralizedTime OPTIONAL,
            endTime  [1] GeneralizedTime OPTIONAL },
            periodic      SET OF Period },
        notThisTime    BOOLEAN DEFAULT FALSE,
        timeZone       TimeZone OPTIONAL } }

Period ::= SEQUENCE {
    timesOfDay [0] SET SIZE (1..MAX) OF DayTimeBand OPTIONAL,
    days       [1] CHOICE {
        intDay      SET OF INTEGER,
        bitDay      BIT STRING { sunday (0),monday (1),tuesday (2),
                               wednesday (3),thursday (4),friday (5),saturday (6) },
        dayOf       XDayOf } OPTIONAL,
    weeks       [2] CHOICE {
        allWeeks    NULL,
        intWeek     SET OF INTEGER,
        dayWeek    }
    }
}

```

```
bitWeek BIT STRING { week1 (0),week2 (1),week3 (2),week4 (3),
week5 (4) } } OPTIONAL,
months [3] CHOICE {
allMonths NULL,
intMonth SET OF INTEGER,
bitMonth BIT STRING { january (0),february (1),march (2),april (3),
may (4),june (5),july (6),august (7),september (8),
october (9),november (10),december (11) }
} OPTIONAL,
Years [4] SET OF INTEGER (1000 .. MAX) OPTIONAL }

XDayOf ::=CHOICE {
first [1] NamedDay,
second [2] NamedDay,
third [3] NamedDay,
fourth [4] NamedDay,
fifth [5] NamedDay ay }

NamedDay ::=CHOICE {
intNamedDays ENUMERATED {
sunday (1),
monday (2),
tuesday (3),
wednesday (4),
thursday (5),
friday (6),
saturday (7) },
bitNamedDays BIT STRING { sunday (0),monday (1),tuesday (2),
wednesday (3),thursday (4),friday (5),saturday (6) } }

DayTimeBand ::=SEQUENCE {
startDayTime [0] DayTime DEFAULT { hour 0 },
endDayTime [1] DayTime DEFAULT { hour 23,minute 59,second 59 } }

DayTime ::=SEQUENCE {
hour [0] INTEGER (0..23),
minute [1] INTEGER (0..59) DEFAULT 0,
second [2] INTEGER (0..59) DEFAULT 0 }

TimeZone ::=INTEGER (-12..12)

TimeAssertion ::=CHOICE {
now NULL,
at GeneralizedTime,
between SEQUENCE {
startTime [0] GeneralizedTime,
endTime [1] GeneralizedTime OPTIONAL,
entirely BOOLEAN DEFAULT FALSE } }

localeContext CONTEXT ::= {
```

```

WITH SYNTAX LocaleContextSyntax
ID           id-avc-locale }

LocaleContextSyntax ::= CHOICE {
    localeID1   OBJECT IDENTIFIER,
    localeID2   DirectoryString {ub-localeContextSyntax} }

ldapAttributeOptionContext CONTEXT ::= {
    WITH SYNTAX      AttributeOptionList
    ASSERTED AS     AttributeOptionList
    ABSENT-MATCH   FALSE
    ID              id-avc-ldapAttributeOption }

AttributeOptionList ::= SEQUENCE OF UTF8String
-- 客体标识符分配 --
-- 在其他模块中分配的客体标识符在解释中说明
-- 属性 --
-- id-at-objectClass          OBJECT IDENTIFIER ::= {id-at 0}
-- id-at-aliasedEntryName      OBJECT IDENTIFIER ::= {id-at 1}
-- id-at-encryptedAliasedEntryName OBJECT IDENTIFIER ::= {id-at 1 2}
id-at-knowledgeInformation   OBJECT IDENTIFIER ::= {id-at 2}
id-at-commonName             OBJECT IDENTIFIER ::= {id-at 3}
-- id-at-encryptedCommonName   OBJECT IDENTIFIER ::= {id-at 3 2}
id-at-surname                OBJECT IDENTIFIER ::= {id-at 4}
-- id-at-encryptedSurname      OBJECT IDENTIFIER ::= {id-at 4 2}
id-at-serialNumber            OBJECT IDENTIFIER ::= {id-at 5}
-- id-at-encryptedSerialNumber OBJECT IDENTIFIER ::= {id-at 5 2}
id-at-countryName            OBJECT IDENTIFIER ::= {id-at 6}
-- id-at-encryptedCountryName  OBJECT IDENTIFIER ::= {id-at 6 2}
id-at-localityName           OBJECT IDENTIFIER ::= {id-at 7}
-- id-at-encryptedLocalityName OBJECT IDENTIFIER ::= {id-at 7 2}
id-at-collectiveLocalityName  OBJECT IDENTIFIER ::= {id-at 7 1}
-- id-at-encryptedCollectiveLocalityName OBJECT IDENTIFIER ::= {id-at 7 1 2}
id-at-stateOrProvinceName    OBJECT IDENTIFIER ::= {id-at 8}
-- id-at-encryptedStateOrProvinceName OBJECT IDENTIFIER ::= {id-at 8 2}
id-at-collectiveStateOrProvinceName OBJECT IDENTIFIER ::= {id-at 8 1}
-- id-at-encryptedCollectiveStateOrProvinceName OBJECT IDENTIFIER ::= {id-at 8 1 2}
id-at-streetAddress           OBJECT IDENTIFIER ::= {id-at 9}
-- id-at-encryptedStreetAddress OBJECT IDENTIFIER ::= {id-at 9 2}
id-at-collectiveStreetAddress  OBJECT IDENTIFIER ::= {id-at 9 1}
-- id-at-encryptedCollectiveStreetAddress OBJECT IDENTIFIER ::= {id-at 9 1 2}
id-at-organizationName        OBJECT IDENTIFIER ::= {id-at 10}
-- id-at-encryptedOrganizationName OBJECT IDENTIFIER ::= {id-at 10 2}
id-at-collectiveOrganizationName OBJECT IDENTIFIER ::= {id-at 10 1}
-- id-at-encryptedCollectiveOrganizationName OBJECT IDENTIFIER ::= {id-at 10 1 2}
id-at-organizationalUnitName  OBJECT IDENTIFIER ::= {id-at 11}

```

-- *id-at-encryptedOrganizationalUnitName*
id-at-collectiveOrganizationalUnitName
-- *id-at-encryptedCollectiveOrganizationalUnitName*
id-at-title
-- *id-at-encryptedTitle*
id-at-description
-- *id-at-encryptedDescription*
id-at-searchGuide
-- *id-at-encryptedSearchGuide*
id-at-businessCategory
-- *id-at-encryptedBusinessCategory*
id-at-postalAddress
-- *id-at-encryptedPostalAddress*
id-at-collectivePostalAddress
-- *id-at-encryptedCollectivePostalAddress*
id-at-postalCode
-- *id-at-encryptedPostalCode*
id-at-collectivePostalCode
-- *id-at-encryptedCollectivePostalCode*
id-at-postOfficeBox
id-at-collectivePostOfficeBox
-- *id-at-encryptedPostOfficeBox*
-- *id-at-encryptedCollectivePostOfficeBox*
id-at-physicalDeliveryOfficeName
id-at-collectivePhysicalDeliveryOfficeName
-- *id-at-encryptedPhysicalDeliveryOfficeName*
-- *id-at-encryptedCollectivePhysicalDeliveryOfficeName*
id-at-telephoneNumber
-- *id-at-encryptedTelephoneNumber*
id-at-collectiveTelephoneNumber
-- *id-at-encryptedCollectiveTelephoneNumber*
id-at-telexNumber
-- *id-at-encryptedTelexNumber*
id-at-collectiveTelexNumber
-- *id-at-encryptedCollectiveTelexNumber*
-- *id-at-teletexTerminalIdentifier*
-- *id-at-encryptedTeletexTerminalIdentifier*
-- *id-at-collectiveTeletexTerminalIdentifier*
-- *id-at-encryptedCollectiveTeletexTerminalIdentifier*
id-at-facsimileTelephoneNumber
-- *id-at-encryptedFacsimileTelephoneNumber*
id-at-collectiveFacsimileTelephoneNumber
-- *id-at-encryptedCollectiveFacsimileTelephoneNumber*

OBJECT IDENTIFIER ::= {id-at 11 2}
OBJECT IDENTIFIER ::= {id-at 11 1}
OBJECT IDENTIFIER ::= {id-at 11 1 2}
 OBJECT IDENTIFIER ::= {id-at 12}
 OBJECT IDENTIFIER ::= {id-at 12 2}
 OBJECT IDENTIFIER ::= {id-at 13}
 OBJECT IDENTIFIER ::= {id-at 13 2}
 OBJECT IDENTIFIER ::= {id-at 14}
 OBJECT IDENTIFIER ::= {id-at 14 2}
 OBJECT IDENTIFIER ::= {id-at 15}
 OBJECT IDENTIFIER ::= {id-at 15 2}
 OBJECT IDENTIFIER ::= {id-at 16}
 OBJECT IDENTIFIER ::= {id-at 16 2}
 OBJECT IDENTIFIER ::= {id-at 16 1}
 OBJECT IDENTIFIER ::= {id-at 16 1 2}
 OBJECT IDENTIFIER ::= {id-at 17}
 OBJECT IDENTIFIER ::= {id-at 17 2}
 OBJECT IDENTIFIER ::= {id-at 17 1}
 OBJECT IDENTIFIER ::= {id-at 17 1 2}
 OBJECT IDENTIFIER ::= {id-at 18}
 OBJECT IDENTIFIER ::= {id-at 18 1}
 OBJECT IDENTIFIER ::= {id-at 18 2}
 OBJECT IDENTIFIER ::= {id-at 18 1 2}
 OBJECT IDENTIFIER ::= {id-at 19}
 OBJECT IDENTIFIER ::= {id-at 19 1}
 OBJECT IDENTIFIER ::= {id-at 19 2}
 OBJECT IDENTIFIER ::= {id-at 19 1 2}
 OBJECT IDENTIFIER ::= {id-at 20}
 OBJECT IDENTIFIER ::= {id-at 20 2}
 OBJECT IDENTIFIER ::= {id-at 20 1}
 OBJECT IDENTIFIER ::= {id-at 20 1 2}
 OBJECT IDENTIFIER ::= {id-at 21}
 OBJECT IDENTIFIER ::= {id-at 21 2}
 OBJECT IDENTIFIER ::= {id-at 21 1}
 OBJECT IDENTIFIER ::= {id-at 21 1 2}
 OBJECT IDENTIFIER ::= {id-at 22}
 OBJECT IDENTIFIER ::= {id-at 22 2}
 OBJECT IDENTIFIER ::= {id-at 22 1}
 OBJECT IDENTIFIER ::= {id-at 22 1 2}
 OBJECT IDENTIFIER ::= {id-at 23}
 OBJECT IDENTIFIER ::= {id-at 23 2}
 OBJECT IDENTIFIER ::= {id-at 23 1}
 OBJECT IDENTIFIER ::= {id-at 23 1 2}

id-at-x121Address
 -- *id-at-encryptedX121Address*
 id-at-internationalISDNNumber
 -- *id-at-encryptedInternationalISDNNumber*
 id-at-collectiveInternationalISDNNumber
 -- *id-at-encryptedCollectiveInternationalISDNNumber*
 id-at-registeredAddress
 -- *id-at-encryptedRegisteredAddress*
 id-at-destinationIndicator
 -- *id-at-encryptedDestinationIndicator*
 id-at-preferredDeliveryMethod
 -- *id-at-encryptedPreferredDeliveryMethod*
 id-at-presentationAddress
 -- *id-at-encryptedPresentationAddress*
 id-at-supportedApplicationContext
 -- *id-at-encryptedSupportedApplicationContext*
 id-at-member
 -- *id-at-encryptedMember*
 id-at-owner
 -- *id-at-encryptedOwner*
 id-at-roleOccupant
 -- *id-at-encryptedRoleOccupant*
 id-at-seeAlso
 -- *id-at-encryptedSeeAlso*
 -- *id-at-userPassword*
 -- *id-at-encryptedUserPassword*
 -- *id-at-userCertificate*
 -- *id-at-encryptedUserCertificate*
 -- *id-at-cACertificate*
 -- *id-at-encryptedCACertificate*
 -- *id-at-authorityRevocationList*
 -- *id-at-encryptedAuthorityRevocationList*
 -- *id-at-certificateRevocationList*
 -- *id-at-encryptedCertificateRevocationList*
 -- *id-at-crossCertificatePair*
 -- *id-at-encryptedCrossCertificatePair*
 id-at-name
 id-at-givenName
 -- *id-at-encryptedGivenName*
 id-at-initials
 -- *id-at-encryptedInitials*
 id-at-generationQualifier
 -- *id-at-encryptedGenerationQualifier*

OBJECT IDENTIFIER ::= {id-at 24}
 OBJECT IDENTIFIER ::= {id-at 24 2}
 OBJECT IDENTIFIER ::= {id-at 25}
 OBJECT IDENTIFIER ::= {id-at 25 2}
 OBJECT IDENTIFIER ::= {id-at 25 1}
 OBJECT IDENTIFIER ::= {id-at 25 1 2}
 OBJECT IDENTIFIER ::= {id-at 26}
 OBJECT IDENTIFIER ::= {id-at 26 2}
 OBJECT IDENTIFIER ::= {id-at 27}
 OBJECT IDENTIFIER ::= {id-at 27 2}
 OBJECT IDENTIFIER ::= {id-at 28}
 OBJECT IDENTIFIER ::= {id-at 28 2}
 OBJECT IDENTIFIER ::= {id-at 29}
 OBJECT IDENTIFIER ::= {id-at 29 2}
 OBJECT IDENTIFIER ::= {id-at 30}
 OBJECT IDENTIFIER ::= {id-at 30 2}
 OBJECT IDENTIFIER ::= {id-at 31}
 OBJECT IDENTIFIER ::= {id-at 31 2}
 OBJECT IDENTIFIER ::= {id-at 32}
 OBJECT IDENTIFIER ::= {id-at 32 2}
 OBJECT IDENTIFIER ::= {id-at 33}
 OBJECT IDENTIFIER ::= {id-at 33 2}
 OBJECT IDENTIFIER ::= {id-at 34}
 OBJECT IDENTIFIER ::= {id-at 34 2}
 OBJECT IDENTIFIER ::= {id-at 35}
 OBJECT IDENTIFIER ::= {id-at 35 2}
 OBJECT IDENTIFIER ::= {id-at 36}
 OBJECT IDENTIFIER ::= {id-at 36 2}
 OBJECT IDENTIFIER ::= {id-at 37}
 OBJECT IDENTIFIER ::= {id-at 37 2}
 OBJECT IDENTIFIER ::= {id-at 38}
 OBJECT IDENTIFIER ::= {id-at 38 2}
 OBJECT IDENTIFIER ::= {id-at 39}
 OBJECT IDENTIFIER ::= {id-at 39 2}
 OBJECT IDENTIFIER ::= {id-at 40}
 OBJECT IDENTIFIER ::= {id-at 40 2}
 OBJECT IDENTIFIER ::= {id-at 41}
 OBJECT IDENTIFIER ::= {id-at 42}
 OBJECT IDENTIFIER ::= {id-at 42 2}
 OBJECT IDENTIFIER ::= {id-at 43}
 OBJECT IDENTIFIER ::= {id-at 43 2}
 OBJECT IDENTIFIER ::= {id-at 44}
 OBJECT IDENTIFIER ::= {id-at 44 2}

id-at-uniqueIdentifier
-- *id-at-encryptedUniqueIdentifier*
id-at-dnQualifier
-- *id-at-encryptedDnQualifier*
id-at-enhancedSearchGuide
-- *id-at-encryptedEnhancedSearchGuide*
id-at-protocolInformation
-- *id-at-encryptedProtocolInformation*
id-at-distinguishedName
-- *id-at-encryptedDistinguishedName*
id-at-uniqueMember
-- *id-at-encryptedUniqueMember*
id-at-houseIdentifier
-- *id-at-encryptedHouseIdentifier*
-- *id-at-supportedAlgorithms*
-- *id-at-encryptedSupportedAlgorithms*
-- *id-at-deltaRevocationList*
-- *id-at-encryptedDeltaRevocationList*
id-at-dmdName
-- *id-at-encryptedDmdName*
-- *id-at-clearance*
-- *id-at-encryptedClearance*
-- *id-at-defaultDirQop*
-- *id-at-encryptedDefaultDirQop*
-- *id-at-attributeIntegrityInfo*
-- *id-at-encryptedAttributeIntegrityInfo*
-- *id-at-attributeCertificate*
-- *id-at-encryptedAttributeCertificate*
-- *id-at-attributeCertificateRevocationList*
-- *id-at-encryptedAttributeCertificateRevocationList*
-- *id-at-confKeyInfo*
-- *id-at-encryptedConfKeyInfo*
-- *id-at-aACertificate*
-- *id-at-attributeDescriptorCertificate*
-- *id-at-attributeAuthorityRevocationList*
-- *id-at-family-information*
id-at-pseudonym
id-at-communicationsService
id-at-communicationsNetwork
-- *id-at-certificationPracticeStmt*
-- *id-at-certificatePolicy*
-- *id-at-pkiPath*
-- *id-at-privPolicy*

OBJECT IDENTIFIER ::= {id-at 45}
OBJECT IDENTIFIER ::= {id-at 45 2}
OBJECT IDENTIFIER ::= {id-at 46}
OBJECT IDENTIFIER ::= {id-at 46 2}
OBJECT IDENTIFIER ::= {id-at 47}
OBJECT IDENTIFIER ::= {id-at 47 2}
OBJECT IDENTIFIER ::= {id-at 48}
OBJECT IDENTIFIER ::= {id-at 48 2}
OBJECT IDENTIFIER ::= {id-at 49}
OBJECT IDENTIFIER ::= {id-at 49 2}
OBJECT IDENTIFIER ::= {id-at 50}
OBJECT IDENTIFIER ::= {id-at 50 2}
OBJECT IDENTIFIER ::= {id-at 51}
OBJECT IDENTIFIER ::= {id-at 51 2}
OBJECT IDENTIFIER ::= {id-at 52}
OBJECT IDENTIFIER ::= {id-at 52 2}
OBJECT IDENTIFIER ::= {id-at 53}
OBJECT IDENTIFIER ::= {id-at 53 2}
OBJECT IDENTIFIER ::= {id-at 54}
OBJECT IDENTIFIER ::= {id-at 54 2}
OBJECT IDENTIFIER ::= {id-at 55}
OBJECT IDENTIFIER ::= {id-at 55 2}
OBJECT IDENTIFIER ::= {id-at 56}
OBJECT IDENTIFIER ::= {id-at 56 2}
OBJECT IDENTIFIER ::= {id-at 57}
OBJECT IDENTIFIER ::= {id-at 57 2}
OBJECT IDENTIFIER ::= {id-at 58}
OBJECT IDENTIFIER ::= {id-at 58 2}
OBJECT IDENTIFIER ::= {id-at 59}
OBJECT IDENTIFIER ::= {id-at 59 2}
OBJECT IDENTIFIER ::= {id-at 60}
OBJECT IDENTIFIER ::= {id-at 60 2}
OBJECT IDENTIFIER ::= {id-at 61}
OBJECT IDENTIFIER ::= {id-at 62}
OBJECT IDENTIFIER ::= {id-at 63}
OBJECT IDENTIFIER {id-at 64}
OBJECT IDENTIFIER ::= {id-at 65}
OBJECT IDENTIFIER ::= {id-at 66}
OBJECT IDENTIFIER ::= {id-at 67}
OBJECT IDENTIFIER ::= {id-at 68}
OBJECT IDENTIFIER ::= {id-at 69}
OBJECT IDENTIFIER ::= {id-at 70}
OBJECT IDENTIFIER ::= {id-at 71}

-- id-at-role	OBJECT IDENTIFIER ::= {id-at 72}
-- id-at-delegationPath	OBJECT IDENTIFIER ::= {id-at 73}
-- id-at-protPrivPolicy	OBJECT IDENTIFIER ::= {id-at 74}
-- id-at-xMLPrivilegeInfo	OBJECT IDENTIFIER ::= {id-at 75}
id-at-uuidpair	OBJECT IDENTIFIER ::= {id-at 76}
-- 控制属性--	
id-cat-sequenceMatchType	OBJECT IDENTIFIER ::= {id-cat 1}
id-cat-wordMatchType	OBJECT IDENTIFIER ::= {id-cat 2}
id-cat-characterMatchTypes	OBJECT IDENTIFIER ::= {id-cat 3}
id-cat-selectedContexts	OBJECT IDENTIFIER ::= {id-cat 4}
-- 通知属性--	
id-not-dSAProblem	OBJECT IDENTIFIER ::= {id-not 0}
id-not-searchServiceProblem	OBJECT IDENTIFIER ::= {id-not 1}
id-not-serviceType	OBJECT IDENTIFIER ::= {id-not 2}
id-not-attributeTypeList	OBJECT IDENTIFIER ::= {id-not 3}
id-not-matchingRuleList	OBJECT IDENTIFIER ::= {id-not 4}
id-not-filterItem	OBJECT IDENTIFIER ::= {id-not 5}
id-not-attributeCombinations	OBJECT IDENTIFIER ::= {id-not 6}
id-not-contextTypeList	OBJECT IDENTIFIER ::= {id-not 7}
id-not-contextList	OBJECT IDENTIFIER ::= {id-not 8}
id-not-contextCombinations	OBJECT IDENTIFIER ::= {id-not 9}
id-not-hierarchySelectList	OBJECT IDENTIFIER ::= {id-not 10}
id-not-searchControlOptionsList	OBJECT IDENTIFIER ::= {id-not 11}
id-not-serviceControlOptionsList	OBJECT IDENTIFIER ::= {id-not 12}
id-not-multipleMatchingLocalities	OBJECT IDENTIFIER ::= {id-not 13}
id-not-proposedRelaxation	OBJECT IDENTIFIER ::= {id-not 14}
id-not-appliedRelaxation	OBJECT IDENTIFIER ::= {id-not 15}
-- 问题定义--	
id-pr-targetDsaUnavailable	OBJECT IDENTIFIER ::= {id-pr 1}
id-pr-dataSourceUnavailable	OBJECT IDENTIFIER ::= {id-pr 2}
id-pr-unidentifiedOperation	OBJECT IDENTIFIER ::= {id-pr 3}
id-pr-unavailableOperation	OBJECT IDENTIFIER ::= {id-pr 4}
id-pr-searchAttributeViolation	OBJECT IDENTIFIER ::= {id-pr 5}
id-pr-searchAttributeCombinationViolation	OBJECT IDENTIFIER ::= {id-pr 6}
id-pr-searchValueNotAllowed	OBJECT IDENTIFIER ::= {id-pr 7}
id-pr-missingSearchAttribute	OBJECT IDENTIFIER ::= {id-pr 8}
id-pr-searchValueViolation	OBJECT IDENTIFIER ::= {id-pr 9}
id-pr-attributeNegationViolation	OBJECT IDENTIFIER ::= {id-pr 10}
id-pr-searchValueRequired	OBJECT IDENTIFIER ::= {id-pr 11}
id-pr-invalidSearchValue	OBJECT IDENTIFIER ::= {id-pr 12}
id-pr-searchContextViolation	OBJECT IDENTIFIER ::= {id-pr 13}
id-pr-searchContextCombinationViolation	OBJECT IDENTIFIER ::= {id-pr 14}
id-pr-missingSearchContext	OBJECT IDENTIFIER ::= {id-pr 15}

id-pr-searchContextValueViolation
id-pr-searchContextValueRequired
id-pr-invalidContextSearchValue
id-pr-unsupportedMatchingRule
id-pr-attributeMatchingViolation
id-pr-unsupportedMatchingUse
id-pr-matchingUseViolation
id-pr-hierarchySelectForbidden
id-pr-invalidHierarchySelect
id-pr-unavailableHierarchySelect
id-pr-invalidSearchControlOptions
id-pr-invalidServiceControlOptions
id-pr-searchSubsetViolation
id-pr-unmatchedKeyAttributes
id-pr-ambiguousKeyAttributes
id-pr-unavailableRelaxationLevel
id-pr-emptyHierarchySelection
id-pr-administratorImposedLimit
id-pr-permanentRestriction
id-pr-temporaryRestriction
id-pr-relaxationNotSupported
-- 匹配规则 --
-- id-mr-objectIdentifierMatch
-- id-mr-distinguishedNameMatch
id-mr-caseIgnoreMatch
id-mr-caseIgnoreOrderingMatch
id-mr-caseIgnoreSubstringsMatch
id-mr-caseExactMatch
id-mr-caseExactOrderingMatch
id-mr-caseExactSubstringsMatch
id-mr-numericStringMatch
id-mr-numericStringOrderingMatch
id-mr-numericStringSubstringsMatch
id-mr-caseIgnoreListMatch
id-mr-caseIgnoreListSubstringsMatch
id-mr-booleanMatch
id-mr-integerMatch
id-mr-integerOrderingMatch
id-mr-bitStringMatch
id-mr-octetStringMatch
id-mr-octetStringOrderingMatch
id-mr-octetStringSubstringsMatch
id-mr-telephoneNumberMatch

OBJECT IDENTIFIER ::= {id-pr 16}
OBJECT IDENTIFIER ::= {id-pr 17}
OBJECT IDENTIFIER ::= {id-pr 18}
OBJECT IDENTIFIER ::= {id-pr 19}
OBJECT IDENTIFIER ::= {id-pr 20}
OBJECT IDENTIFIER ::= {id-pr 21}
OBJECT IDENTIFIER ::= {id-pr 22}
OBJECT IDENTIFIER ::= {id-pr 23}
OBJECT IDENTIFIER ::= {id-pr 24}
OBJECT IDENTIFIER ::= {id-pr 25}
OBJECT IDENTIFIER ::= {id-pr 26}
OBJECT IDENTIFIER ::= {id-pr 27}
OBJECT IDENTIFIER ::= {id-pr 28}
OBJECT IDENTIFIER ::= {id-pr 29}
OBJECT IDENTIFIER ::= {id-pr 30}
OBJECT IDENTIFIER ::= {id-pr 31}
OBJECT IDENTIFIER ::= {id-pr 32}
OBJECT IDENTIFIER ::= {id-pr 33}
OBJECT IDENTIFIER ::= {id-pr 34}
OBJECT IDENTIFIER ::= {id-pr 35}
OBJECT IDENTIFIER ::= {id-pr 36}

OBJECT IDENTIFIER ::= {id-mr 0}
OBJECT IDENTIFIER ::= {id-mr 1}
OBJECT IDENTIFIER ::= {id-mr 2}
OBJECT IDENTIFIER ::= {id-mr 3}
OBJECT IDENTIFIER ::= {id-mr 4}
OBJECT IDENTIFIER ::= {id-mr 5}
OBJECT IDENTIFIER ::= {id-mr 6}
OBJECT IDENTIFIER ::= {id-mr 7}
OBJECT IDENTIFIER ::= {id-mr 8}
OBJECT IDENTIFIER ::= {id-mr 9}
OBJECT IDENTIFIER ::= {id-mr 10}
OBJECT IDENTIFIER ::= {id-mr 11}
OBJECT IDENTIFIER ::= {id-mr 12}
OBJECT IDENTIFIER ::= {id-mr 13}
OBJECT IDENTIFIER ::= {id-mr 14}
OBJECT IDENTIFIER ::= {id-mr 15}
OBJECT IDENTIFIER ::= {id-mr 16}
OBJECT IDENTIFIER ::= {id-mr 17}
OBJECT IDENTIFIER ::= {id-mr 18}
OBJECT IDENTIFIER ::= {id-mr 19}
OBJECT IDENTIFIER ::= {id-mr 20}

id-mr-telephoneNumberSubstringsMatch
 id-mr-presentationAddressMatch
 id-mr-uniqueMemberMatch
 id-mr-protocolInformationMatch
 id-mr-uTCTimeMatch
 id-mr-uTCTimeOrderingMatch
 id-mr-generalizedTimeMatch
 id-mr-generalizedTimeOrderingMatch
 id-mr-integerFirstComponentMatch
 id-mr-objectIdentifierFirstComponentMatch
 id-mr-directoryStringFirstComponentMatch
 id-mr-wordMatch
 id-mr-keywordMatch
 -- *id-mr-certificateExactMatch*
 -- *id-mr-certificateMatch*
 -- *id-mr-certificatePairExactMatch*
 -- *id-mr-certificatePairMatch*
 -- *id-mr-certificateListExactMatch*
 -- *id-mr-certificateListMatch*
 -- *id-mr-algorithmIdentifierMatch*
 id-mr-storedPrefixMatch
 -- *id-mr-attributeCertificateMatch*
 -- *id-mr-readerAndKeyIDMatch*
 -- *id-mr-attributeIntegrityMatch*
 废止
 -- *id-mr-attributeCertificateExactMatch*
 -- *id-mr-holderIssuerMatch*
 id-mr-systemProposedMatch
 id-mr-generalWordMatch
 id-mr-approximateStringMatch
 id-mr-ignoreIfAbsentMatch
 id-mr-nullMatch
 id-mr-zonalMatch
 -- *id-mr-authAttIdMatch*
 -- *id-mr-roleSpecCertIdMatch*
 -- *id-mr-basicAttConstraintsMatch*
 -- *id-mr-delegatedNameConstraintsMatch*
 -- *id-mr-timeSpecMatch*
 -- *id-mr-attDescriptorMatch*
 -- *id-mr-acceptableCertPoliciesMatch*
 -- *id-mr-policyMatch*
 -- *id-mr-delegationPathMatch*
 -- *id-mr-pkiPathMatch*

OBJECT IDENTIFIER ::= {id-mr 21}
 OBJECT IDENTIFIER ::= {id-mr 22}
 OBJECT IDENTIFIER ::= {id-mr 23}
 OBJECT IDENTIFIER ::= {id-mr 24}
 OBJECT IDENTIFIER ::= {id-mr 25}
 OBJECT IDENTIFIER ::= {id-mr 26}
 OBJECT IDENTIFIER ::= {id-mr 27}
 OBJECT IDENTIFIER ::= {id-mr 28}
 OBJECT IDENTIFIER ::= {id-mr 29}
 OBJECT IDENTIFIER ::= {id-mr 30}
 OBJECT IDENTIFIER ::= {id-mr 31}
 OBJECT IDENTIFIER ::= {id-mr 32}
 OBJECT IDENTIFIER ::= {id-mr 33}
 OBJECT IDENTIFIER ::= {id-mr 34}
 OBJECT IDENTIFIER ::= {id-mr 35}
 OBJECT IDENTIFIER ::= {id-mr 36}
 OBJECT IDENTIFIER ::= {id-mr 37}
 OBJECT IDENTIFIER ::= {id-mr 38}
 OBJECT IDENTIFIER ::= {id-mr 39}
 OBJECT IDENTIFIER ::= {id-mr 40}
 OBJECT IDENTIFIER ::= {id-mr 41}
 OBJECT IDENTIFIER ::= {id-mr 42}
 OBJECT IDENTIFIER ::= {id-mr 43}
 OBJECT IDENTIFIER ::= {id-mr 44}
 OBJECT IDENTIFIER ::= {id-mr 45}
 OBJECT IDENTIFIER ::= {id-mr 46}
 OBJECT IDENTIFIER ::= {id-mr 47}
 OBJECT IDENTIFIER ::= {id-mr 48}
 OBJECT IDENTIFIER ::= {id-mr 49}
 OBJECT IDENTIFIER ::= {id-mr 50}
 OBJECT IDENTIFIER ::= {id-mr 51}
 OBJECT IDENTIFIER ::= {id-mr 52}
 OBJECT IDENTIFIER ::= {id-mr 53}
 OBJECT IDENTIFIER ::= {id-mr 54}
 OBJECT IDENTIFIER ::= {id-mr 55}
 OBJECT IDENTIFIER ::= {id-mr 56}
 OBJECT IDENTIFIER ::= {id-mr 57}
 OBJECT IDENTIFIER ::= {id-mr 58}
 OBJECT IDENTIFIER ::= {id-mr 59}
 OBJECT IDENTIFIER ::= {id-mr 60}
 OBJECT IDENTIFIER ::= {id-mr 61}
 OBJECT IDENTIFIER ::= {id-mr 62}

id-mr-facsimileNumberMatch	OBJECT IDENTIFIER ::= {id-mr 63}
id-mr-facsimileNumberSubstringsMatch	OBJECT IDENTIFIER ::= {id-mr 64}
-- id-mr-enhancedCertificateMatch	OBJECT IDENTIFIER ::= {id-mr 65}
-- id-mr-sOAIdentifierMatch	OBJECT IDENTIFIER ::= {id-mr 66}
-- id-mr-indirectIssuerMatch	OBJECT IDENTIFIER ::= {id-mr 67}
id-mr-uuidpairmatch	OBJECT IDENTIFIER ::= {id-mr 68}
-- 上下文 --	
id-avc-language	OBJECT IDENTIFIER ::= {id-avc 0}
id-avc-temporal	OBJECT IDENTIFIER ::= {id-avc 1}
id-avc-locale	OBJECT IDENTIFIER ::= {id-avc 2}
-- id-avc-attributeValueSecurityLabelContext	OBJECT IDENTIFIER ::= {id-avc 3}
-- id-avc-attributeValueIntegrityInfoContext	OBJECT IDENTIFIER ::= {id-avc 4}
id-avc-ldapAttributeOption	OBJECT IDENTIFIER ::= {id-avc 5}
END -- SelectedAttributeTypes	

国家图书馆专用

附录 B
(资料性附录)
属性类型概要

本附录总结了在本目录规范中参考或规定的所选属性类型并示出了它们的层次关系。共用一个通常 ASN.1 语法的属性在该语法下示出为缩进,且其他属性的图表类型属性在其父型下示出为缩进。属于相关非集合属性的子类型的集合属性不表示,但是相关属性用一个星号(*)标记。通知属性用一个符号(#)标记。

<i>DirectoryString</i>		
name		member
commonName		owner
surname		roleOccupant
givenName		seeAlso
initials	<i>FilterItem</i>	
generationQualifier	filterItem #	
countryName	<i>AttributeCombination</i>	
localityName *	attributeCombinations #	
stateOrProvinceName *	<i>ContextAssertion</i>	
organizationName *	contextList #	
organizationalUnitName *	<i>ContextCombination</i>	
pseudonym	contextCombinations #	
title	<i>HierarchySelections</i>	
dmdName	hierarchySelectList #	
streetAddress *	<i>SearchControlOptions</i>	
houseIdentifier	searchControlOptionsList #	
description	<i>ServiceControlOptions</i>	
businessCategory		
postalCode *	<i>MultipleMatchingLocalities</i>	
postOfficeBox *		
physicalDeliveryOfficeName *		
serviceControlOptionsList #		
knowledgeInformation		
multipleMatchingLocalities		
<i>PrintableString</i>		
serialNumber	<i>MRMappings</i>	
dnQualifier	proposedRelaxation	
destinationIndicator		
telephoneNumber *	<i>Guide</i>	
	searchGuide	

NumericString

x121Address
internationalISDNNumber *

EnhancedGuide

enhancedSearchGuide

OBJECT IDENTIFIER

communicationsService
communicationsNetwork
supportedApplicationContext
dSAPproblem #
searchServiceProblem #
serviceType #
attributeTypeList #
facsimileTelephoneNumber *
matchingRuleList #
contextTypeList #
appliedRelaxation #

PostalAddress

postalAddress *
registeredAddress

TelexNumber

telexNumber *

FacsimileTelephoneNumber

PresentationAddress

presentationAddress

BIT STRING

uniqueIdentifier

ProtocolInformation

protocolInformation

NameAndOptionalUID

uniqueMember

PreferredDeliveryMethod

preferredDeliveryMethod

DistinguishedName

distinguishedName

UUIDPair

UUIDPair

附录 C
(资料性附录)
上界

本附录包括在本目录规范中使用的所有建议的上界值,形式为 ASN.1 模块UpperBounds。

UpperBounds {joint-iso-itu-t ds(5) module(1) upperBounds(10) 5}

DEFINITIONS ::=

BEGIN

-- EXPORTS All --

-- 本模块中定义的类型和值输出可用于目录规范包含的其他 ASN.1 模块,
-- 以及使用它们访问目录服务的其他应用。
-- 其他应用可把它们用于自己的目的,
-- 但这并不限制为维护或改进目录服务所需的扩充和修改。

ub-answerback	INTEGER	::=8
ub-business-category	INTEGER	::=128
ub-common-name	INTEGER	::=64
ub-content	INTEGER	::=32768
ub-country-code	INTEGER	::=4
ub-description	INTEGER	::=1024
ub-destination-indicator	INTEGER	::=128
ub-directory-string-first-component-match	INTEGER	::=32768
ub-domainLocalID	INTEGER	::=64
ub-international-isdn-number	INTEGER	::=16
ub-knowledge-information	INTEGER	::=32768
ub-labeledURI	INTEGER	::=32768
ub-localeContextSyntax	INTEGER	::=128
ub-locality-name	INTEGER	::=128
ub-match	INTEGER	::=128
ub-name	INTEGER	::=64
ub-organization-name	INTEGER	::=64
ub-organizational-unit-name	INTEGER	::=64
ub-physical-office-name	INTEGER	::=128
ub-post-office-box	INTEGER	::=40
ub-postal-code	INTEGER	::=40
ub-postal-line	INTEGER	::=6
ub-postal-string	INTEGER	::=30
ub-privacy-mark-length	INTEGER	::=128
ub-pseudonym	INTEGER	::=128
ub-saslMechanism	INTEGER	::=64
ub-schema	INTEGER	::=1024
ub-search	INTEGER	::=32768
ub-serial-number	INTEGER	::=64

ub-state-name	INTEGER	::=128
ub-street-address	INTEGER	::=128
ub-surname	INTEGER	::=64
ub-tag	INTEGER	::=64
ub-telephone-number	INTEGER	::=32
ub-teletex-terminal-id	INTEGER	::=1024
ub-telex-number	INTEGER	::=14
ub-title	INTEGER	::=64
ub-user-password	INTEGER	::=128
ub-x121-address	INTEGER	::=15
END -- 上界		

国家图书馆专用

附录 D
(资料性附录)

属性、匹配规则和上下文的英文字母表顺序索引

本附录按英文字母表顺序列出了在本目录规范中规定的所有属性和匹配规则以及与规定它们的章节之间的相互参照。

应用张弛 applied relaxation	5.12.16
近似串匹配 approximate string match	7.6.1
属性组合 attribute combinations	5.12.7
属性类型列表 attribute type list	5.12.4
比特串匹配 bit string match	7.2.4
布尔匹配 boolean match	7.2.1
商业类别 business category	5.5.4
大小写精确匹配 case exact match	7.1.1
大小写精确排序匹配 case exact ordering match	7.1.2
大小写精确子串匹配 case exact substrings match	7.1.3
不区分大小写的列表匹配 case ignore list match	7.1.7
不区分大小写的列表子串匹配 case ignore list substrings match	7.1.8
不区分大小写的匹配 case ignore match	7.1.1
不区分大小写的排序匹配 case ignore ordering match	7.1.2
不区分大小写的子串匹配 case ignore substrings match	7.1.3
公共名(称) common name	5.2.2
通信网络 communications network	5.7.10
通信业务 communications service	5.7.9
组件匹配 component match	7.2.16
上下文组合 context combinations	5.12.10
上下文列表 context list	5.12.9
上下文类型列表 context type list	5.12.8
国家名 country name	5.3.1
描述 description	5.5.1
目的地指示符 destination indicator	5.7.8
目录串第一组件匹配 directory string first component match	7.4.3
可辨别名 distinguished name	5.10.1
DMD 名 DMD name	5.11.1
可辨别名限定符 DN qualifier	5.2.8
DSA 问题 DSA problem	5.12.1
增强的搜索指南 enhanced search guide	5.5.3
传真号码匹配 facsimile number match	7.2.13
传真号码子串匹配 facsimile number substrings match	7.2.14
传真电话号码 facsimile telephone number	5.7.4
过滤项 filter item	5.12.6
通用字匹配 general word match	7.5.3

代限定符 generation qualifier	5.2.6
通用时间匹配 generalized time match	7.3.3
通用时间排序匹配 generalized time ordering match	7.3.4
给定名 given name	5.2.4
层次选择列表 hierarchy select list	5.12.11
住宅标识符 house identifier	5.3.5
不存在则忽略匹配 ignore if absent match	7.7.1
首字母 initials	5.2.5
整数第一组件匹配 integer first component match	7.4.1
整数匹配 integer match	7.2.2
整数排序匹配 integer ordering match	7.2.3
国际 ISDN 号码 international ISDN number	5.7.6
关键字匹配 keyword match	7.5.2
知识信息 knowledge information	5.1.1
语言上下文 language context	8.1
LDAP 属性选项上下文 LDAP attribute option context	8.4
场所上下文 locale context	8.3
地点名 locality name	5.3.2
匹配规则列表 matching rule list	5.12.5
成员 member	5.10.2
多个匹配的地理位置 multiple matching localities	5.12.14
名(称) name	5.2.1
空匹配 null match	7.7.2
数字串匹配 numeric string match	7.1.4
数字串排序匹配 numeric string ordering match	7.1.5
数字串子串匹配 numeric string substrings match	7.1.6
客体标识符第一组件匹配 object identifier first component match	7.4.2
八位位组串匹配 octet string match	7.2.5
八位位组串排序匹配 octet string ordering match	7.2.6
八位位组串子串匹配 octet string substrings match	7.2.7
组织单元名 organizational unit name	5.4.2
组织名 organization name	5.4.1
所有者 owner	5.10.4
物理投递邮局名 physical delivery office name	5.6.4
邮政信箱 post office box	5.6.3
邮政地址 postal address	5.6.1
邮政编码 postal code	5.6.2
优选投递方式 preferred delivery method	5.8.1
表示地址 presentation address	5.9.1
表示地址匹配 presentation address match	7.2.10
建议张弛 proposed relaxation	5.12.15
协议信息 protocol information	5.9.3
协议信息匹配 protocol information match	7.2.12

假名	pseudonym	5.2.10
注册地址	registered address	5.7.7
角色承担者	role occupant	5.10.5
搜索指南	search guide	5.5.2
搜索控制选项列表	search control options list	5.12.12
搜索服务问题	search service problem	5.12.2
另见	see also	5.10.6
序列号	serial number	5.2.9
服务控制选项列表	service control options list	5.12.13
服务类型	service type	5.12.3
省或州名	state or province name	5.3.3
存储前缀匹配	stored prefix match	7.1.9
街道地址	street address	5.3.4
所支持的应用上下文	supported application context	5.9.2
姓	surname	5.2.3
系统提议的匹配	system proposed match	7.3.5
电话号码	telephone number	5.7.1
电话号码匹配	telephone number match	7.2.8
电话号码子串匹配	telephone number substrings match	7.2.9
智能用户电报终端标识符	teletex terminal identifier	5.7.3
电报号码	telex number	5.7.2
时间上下文	temporal context	8.2
头衔	title	5.4.3
唯一标识符	unique identifier	5.2.7
唯一成员	unique member	5.10.3
唯一成员匹配	unique member match	7.2.11
全球唯一标识符对	universal unique identifier pair	5.2.11
UTC 时间匹配	UTC time match	7.3.1
UTC 时间排序匹配	UTC time ordering match	7.3.2
UUID 对匹配	UUID pair match	7.2.15
字匹配	word match	7.5.1
X.121 地址	X.121 address	5.7.5
区匹配	zonal match	7.8

国家图书馆专用

附录 E
(资料性附录)
区匹配规则的示例

注：下述解释对本目录规范的 7.8 定义的区匹配给出了相关的示例。为了有助于明确这些示例所应用的情况，保留了确定的文本，但使用斜体字表示。

在区匹配中，核心机制实现了一个映射，从搜索操作的过滤器中使用的一个串断言或断言的组合，映射为一个不可再分的特性的集合，这些特性可能被客体所拥有，且在相应条目中通过属性所描述。这种映射通过一个可替代的过滤项的集合来表示，这些过滤项替代了原始过滤器中的过滤项。用于在过滤器中表示断言的属性不必与条目中用于表示客体特性的属性相同。这里说明了一个具体的区匹配是如何发生的：

- 某个用户搜索一个名为史密斯先生的电话客户，他住在布拉克内尔(Bracknell)，该客户使用了一个过滤器： $\{\{\text{locality}=\text{Bracknell}\} \text{ AND } \{\text{surname}=\text{Smithers}\}\}$ 。
- 目录包含一个地理方面的映射(被称为一个地名词典)，该词典将 Bracknell 映射为在 Bracknell 区域内服务的几个区的邮政编码(例如 RG12 2JL)，结果是将过滤器转换为： $\{\{\text{zone}=\text{b1}\} \text{ OR } \{\text{zone}=\text{b2}\} \dots\} \text{ AND } \{\text{surname}=\text{Smithers}\}$ 。这里 b1,b2,...,bn 是表示 Bracknell 的邮政编码集；每个不同的住处都有一个单独的邮政编码，而一个大的建筑物或场所可能会有多个。目前，匹配试图定位某个给定了姓的人，其地理位置共享一个由 b1 或 b2 等给定的一个公共区域。
- 如果搜索不成功，则映射将自动放宽以包含更多的区(即临近的邮政编码)；这就可能会发现一个名为史密斯的客户，他住在 Newell Green 镇(该镇紧挨着 Bracknell)。

一个基于映射的匹配规则可以使用替代名(称)和冗余的信息，并且可以组合多种断言，例如，

$\{\{\text{locality}=\text{Newton}\} \text{ AND } \{\text{locality}=\text{Cumbria}\}\}$ ；它甚至能够标识同一个断言中的多个组件，例如， $\{\text{locality}=\text{"Newton, Cumbria"}\}$ 。因此，本例中的匹配还可能为下述情况服务：

- $\{\{\text{locality}=\text{Bullbrook}\} \text{ AND } \{\text{surname}=\text{Smithers}\}\}$ (这里 Bullbrook 是 Bracknell 内的一个区)
- $\{\{\text{locality}=\text{Bracknell}\} \text{ AND } \{\text{locality}=\text{Bullbrook}\} \text{ AND } \{\text{surname}=\text{Smithers}\}\}$
- $\{\{\text{locality}=\text{Bullbrook, Bracknell}\} \text{ AND } \{\text{surname}=\text{Smithers}\}\}$
- $\{\text{locality}=\text{Berks}\} \text{ AND } \{\text{locality}=\text{Bracknell}\} \text{ AND } \{\text{locality}=\text{Bullbrook}\} \text{ AND } \{\text{surname}=\text{Smithers}\}\}$ (Bracknell 处于 Berkshire 的老城边界内，Berkshire 简写为 Berks)
- $\{\{\text{locality}=\text{Berkshire}\} \text{ AND } \{\text{locality}=\text{Bracknell}\} \text{ AND } \{\text{locality}=\text{Bullbrook}\} \text{ AND } \{\text{surname}=\text{Smithers}\}\}$
- $\{\{\text{locality}=\text{East Berks}\} \text{ AND } \{\text{locality}=\text{Bracknell Forest}\} \text{ AND } \{\text{surname}=\text{Smithers}\}\}$ (Bracknell 所处的新的地区管理区被称为 East Berks；本地管理行政区被称为 Bracknell Forest)
- $\{\{\text{postcode}=\text{RG12 2JL}\} \text{ AND } \{\text{surname}=\text{Smithers}\}\}$ (RG12 2JL 是 Bullbrook 的 20 个左右的邮政编码之一)

区匹配规则是与地理匹配相关的基于映射的匹配规则。它们基于一个关于地方名(称)的词典，即地名词典。一个地名词典一般来说会涵盖(即提供一个相关的地理数据库)组成一个国家或区域的范围。一个地理方面的搜索请求必须根据某个具体的地名词典来进行解释。一个地名词典主要是将表示地方名(称)的串与某个已命名地方相关联起来，已命名地方由一个或多个表示地方名(称)的串来标识。例如，在大不列颠，由地方名(称)串所标识的已命名地方的示例有：Devon 的“Mogworthy”、Hertfordshire 的“Offleyhoo”、“Thames Valley”以及“London”等。

某些地方名(称)串可以直接映射为一个单独的已命名地方，但这并不总是可能的。不能够明确标识地方的地方名(称)示例有：“Newton”，“Lees”等，因为每个这样的名(称)都对应多个已命名地方。因此，一个已命名地方需要被多个不同的地方名(称)来标识；例如，下述是 3 个已命名地方：“Newton”

“Tattenhall”“Cheshire”), (“Newton”“Chester”“Cheshire”) 和 (“Newton”“Cumbria”), 其中每个地方名(称)的组合由圆括号所指示。

一个地方名(称)本身内部可能就有多个组件, 例如 “London Heathrow”、“Newton Abbott” 等, 但每个地方名(称)都被认为是一个单独的串, 因为即使在本地, 如果不包括全部的组件, 这些名(称)就不是完整的; 或者因为一个组件(如 “Abbott”) 在语义上就不是一个地方名(称)(在标准的地名词典中, 没有一个地方名(称)会被赋值为 “Abbott”)。一个已命名地方还可能通过它的多个名(称)中的一个子集来标识; 例如 (“Newton”“Tattenhall”) 可以充分地定义之前所提到的一个地方。然而, 在这种情况下, 与仅需要国家名来进行限定的 Newtons, 如 (“Newton”“Cumbria”) 来进行类比, (“Newton”“Tattenhall”“Cheshire”) 可能是更有用的一个组合。

下面是对区匹配所隐含的模型的更正式的陈述:

- a) 区匹配的基础是存在一个或多个地名词典, 这些地名词典由 DSA 出于某种目的而支持。一个地名词典是一个地理方面的字典, 由一个适当的数据库所支持, 在它范围内覆盖了一个国家或一个已命名地区。对于一个具体的搜索, 对域的选择是通过本地方式来执行的。例如, 一个地名词典能够涵盖英国大陆(包括英格兰、苏格兰和威尔士)以及一些边远的岛屿。地名词典中包含了地方名(称)以及它们的特性, 并列出了与之相匹配的已命名地方。这是由地方名(称)特性的查找和排序比较机制来支持的, 这些特性由可组合的位置属性给出, 且完全独立于 DIT。在图 E. 1 中, 区域的轮廓由一条粗线所标识;
- b) 一个地名词典所涵盖的区域中包括地方。图 E. 1 中, 区域的轮廓由相应于文字的边界所标识。一个地方是一个被公认的已命名地理区域; 地方之间可以重叠, 甚至可以扩展至稍微超出区域的边界之外(如图 E. 1 中的 F)。地方的示例有: England、Berkshire、Bracknell、Bullbrook(这四个地方是逐次嵌套的)以及 Thames Valley(该地方包含了 Berkshire 的部分, 但又超出了它的范围)。通过参考地名词典而可标识的地方被称为“已命名地方”;
- c) 地名词典本身是基于那些表示地方名(称)的串(例如 “England”、“Berkshire”、“Bracknell”、“Bullbrook”、“Thames Valley”等)。这些串用来标识(或命名)已命名地方。一个已命名地方的名(称)可以是:
 - 一个单独的地方名(称), 可能由多个单词组成, 如 “Newton Abbott”;
 - 多个地方名(称)的一个集合, 一般来说, 一个地方名(称)相应于一个较大的地区(如 “Cumbria”), 通过该集合可以(在上下文内)将地方名(称)限定到一个相应的较小地区(如 “Newton”)。

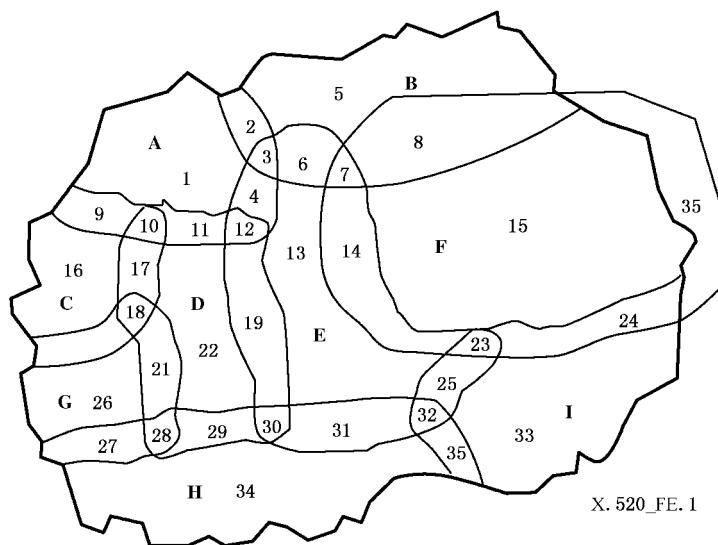


图 E. 1 区域、地方和区

一般来说,一个已命名地方在地名词典内必须与容纳该地方的一个较大范围的名(称)相关联,即使不要求唯一的标识符。例如,地名词典需要定义可访问的 Newton Abbott 城为“Newton Abbott”,或者为(“Newton Abbott”“Devon”),这样就与地方名(称)“Devon”相关联起来(该名(称)是“Devonshire”的同义词)。

国家图书馆专用

国家图书馆专用

中华人民共和国
国家标准
信息技术 开放系统互连 目录
第6部分：选定的属性类型

GB/T 16264. 6—2008/ISO/IEC 9594-6:2005

*

中国标准出版社出版发行
北京复兴门外三里河北街16号

邮政编码：100045

网址 www.spc.net.cn

电话：68523946 68517548

中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

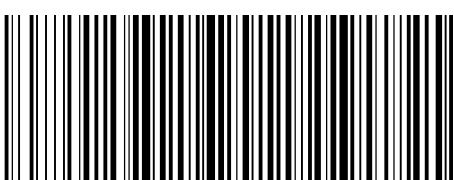
*

开本 880×1230 1/16 印张 5.25 字数 154 千字
2008年11月第一版 2008年11月第一次印刷

*

书号：155066 · 1-34677

如有印装差错 由本社发行中心调换
版权所有 侵权必究
举报电话：(010)68533533



GB/T 16264. 6-2008