国 际 电 信 联 盟

ITU-T X.511

国际电信联盟
电信标准化部门

(08/2005)

X系列：数据网、开放系统通信和安全性

号码簿

信息技术 — 开放系统互连 — 号码簿：抽象服务定义

ITU-T X.511建议书

国际电信联盟

ITU-T X系列建议书

数据网、开放系统通信和安全性

公众数据网	
服务和设施	X.1-X.19
接口	X.20-X.49
传输、信令和交换	X.50-X.89
网络概貌	X.90-X.149
维护	X.150-X.179
管理安排	X.180-X.199
开放系统互连	
模型和记法	X.200-X.209
服务限定	X.210-X.219
连接式协议规范	X.220-X.229
无连接式协议规范	X.230-X.239
PICS 书写形式	X.240-X.259
协议标识	X.260-X.269
安全协议	X.270-X.279
层管理对象	X.280-X.289
一致性测试	X.290-X.299
网间互通	
概述	X.300-X.349
卫星数据传输系统	X.350-X.369
以IP为基础的网络	X.370-X.379
报文处理系统	X.400-X.499
号码簿	**X.500-X.599**
OSI组网和系统概貌	
组网	X.600-X.629
效率	X.630-X.639
服务质量	X.640-X.649
命名、寻址和登记	X.650-X.679
抽象句法记法1（ASN.1）	X.680-X.699
OSI管理	
系统管理框架和结构	X.700-X.709
管理通信服务和协议	X.710-X.719
管理信息的结构	X.720-X.729
管理功能	X.730-X.799
安全	X.800-X.849
OSI应用	
托付、并发和恢复	X.850-X.859
事务处理	X.860-X.879
远程操作	X.880-X.889
ASN.1的一般应用	X.890-X.899
开放分布式处理	X.900-X.999
电信安全	X.1000-

欲了解更详细信息，请查阅*ITU-T*建议书目录。

国际标准ISO/IEC 9594-3
ITU-T X.511建议书

信息技术 — 开放系统互连 — 号码簿：抽象服务定义

摘　要

本建议书|国际标准以一种抽象的方式定义了号码簿提供的、外形上可见的服务，包括绑定的和解开的操作、读操作、搜索操作、修改操作和错误。

来　源

ITU-T 第 17 研究组（2005-2008）按照 ITU-T A.8 建议书规定的程序，于 2005 年 8 月 29 日批准了 ITU-T X.511 建议书。等同的文本也作为 ISO/IEC 9594-3 出版。

前　言

国际电信联盟（ITU）是从事电信领域工作的联合国专门机构。ITU-T（国际电信联盟电信标准化部门）是国际电信联盟的常设机构，负责研究技术、操作和资费问题，并且为在世界范围内实现电信标准化，发表有关上述研究项目的建议书。

每四年一届的世界电信标准化全会（WTSA）确定 ITU-T 各研究组的研究课题，再由各研究组制定有关这些课题的建议书。

WTSA 第 1 号决议规定了批准建议书须遵循的程序。

属 ITU-T 研究范围的某些信息技术领域的必要标准，是与国际标准化组织（ISO）和国际电工技术委员会（IEC）合作制定的。

注

本建议书为简明扼要起见而使用的“主管部门”一词，既指电信主管部门，又指经认可的运营机构。

遵守本建议书的规定是以自愿为基础的，但建议书可能包含某些强制性条款（以确保例如互操作性或适用性等），只有满足所有强制性条款的规定，才能达到遵守建议书的目的。“应该”或“必须”等其它一些强制性用语及其否定形式被用于表达特定要求。使用此类用语不表示要求任何一方遵守本建议书。

知识产权

国际电联提请注意：本建议书的应用或实施可能涉及使用已申报的知识产权。国际电联对无论是其成员还是建议书制定程序之外的其它机构提出的有关已申报的知识产权的证据、有效性或适用性不表示意见。

至本建议书批准之日止，国际电联尚未收到实施本建议书可能需要的受专利保护的知识产权的通知。但需要提醒实施者注意的是，这可能并非最新信息，因此特大力提倡他们通过下列网址查询电信标准化局（TSB）的专利数据库：http://www.itu.int/ITU-T/ipr/。

目 录

页

引言

本建议书|国际标准连同本系列其他建议书|国际标准是为方便信息处理系统之间的互连以提供号码簿服务而制定的。所有这些系统的集合，连同它们所拥有的号码簿信息可被视为一个整体，被称为**号码簿**。号码簿所拥有的信息，总称为号码簿信息库（DIB），典型地被用于方便对象之间的通信、与对象的通信或有关对象的通信等，这些对象如应用实体、个人、终端和分发表等。

号码簿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。

本建议书 | 国际标准定义了号码簿为其用户提供的性能。

本建议书|国际标准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。目前本建议书|国际标准的第 5 版是原有第 4 版的修订和增强，但不是替代。在系统实现时仍可以声明为遵循第 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来的错误）。建议在系统实现时尽快遵循第 5 版。

第 5 版详细定义了号码簿协议的第 1 版和第 2 版。

第 1 版和第 2 版仅定义了协议第 1 版。本版本（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号码簿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 ITU-T X.519 建议书| ISO/IEC 9594-5 中定义的扩展规则调节。

附件 A 是本建议书 | 国际标准的组成部分，提供了号码簿抽象服务的 ASN.1 模块。

附件 B 不是本建议书 | 国际标准的组成部分，提供了用于描述与基本访问控制相关的语义的图表，它适用于号码簿操作的处理。

附件 C 不是本建议书 | 国际标准的组成部分，给出了条目族使用的例子。

附件 D 不是本建议书 | 国际标准的组成部分，列出了已纳入并形成本建议书 | 国际标准该版本的修正和缺陷报告。

国际标准
ITU-T 建议书

信息技术 — 开放系统互连 — 号码簿：抽象服务定义

1 范围

本建议书 | 国际标准以一种抽象的方式定义了号码簿提供的、外形上可见的服务。

本建议书 | 国际标准不规定单个实现或产品。

2 参考文献

下列建议书和国际标准的条款，在本建议书中的引用而构成本建议书 | 国际标准的条款。在出版时，所指出的版本是有效的。所有的建议书和国际标准均会得到修订，本建议书 | 国际标准的使用者应查证是否有可能使用下列建议书和国际标准最新版本。IEC 和 ISO 的各成员有目前有效的国际标准的目录。国际电联电信标准化局有目前有效的 ITU-T 建议书的清单。

2.1 等同的建议书 | 国际标准

- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*.
- ITU-T Recommendation X.500 (2005) | ISO/IEC 9594-1:2005, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services*.
- ITU-T Recommendation X.501 (2005) | ISO/IEC 9594-2:2005, *Information technology – Open Systems Interconnection – The Directory: Models*.
- ITU-T Recommendation X.509 (2005) | ISO/IEC 9594-8:2005, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks.*
- ITU-T Recommendation X.518 (2005) | ISO/IEC 9594-4:2005, *Information technology – Open Systems Interconnection – The Directory: Procedures for distributed operation.*
- ITU-T Recommendation X.519 (2005) | ISO/IEC 9594-5:2005, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications.*
- ITU-T Recommendation X.520 (2005) | ISO/IEC 9594-6:2005, *Information technology – Open Systems Interconnection – The Directory: Selected attribute types.*
- ITU-T Recommendation X.521 (2005) | ISO/IEC 9594-7:2005, *Information technology – Open Systems Interconnection – The Directory: Selected object classes.*
- ITU-T Recommendation X.525 (2005) | ISO/IEC 9594-9:2005, *Information technology – Open Systems Interconnection – The Directory: Replication.*
- ITU-T Recommendation X.530 (2005) | ISO/IEC 9594-10:2005, *Information technology – Open Systems Interconnection – The Directory: Use of systems management for administration of the Directory*.
- ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.681 (2002) | ISO/IEC 8824-2:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- ITU-T Recommendation X.682 (2002) | ISO/IEC 8824-3:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*

– ITU-T Recommendation X.683 (2002) | ISO/IEC 8824-4:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*

2.2 其他参考文献

– RFC 2025 (1996), *The Simple Public-Key GSS-API Mechanism (SPKM).*

– RFC 2222 (1997), *Simple Authentication and Security Layer (SASL).*

3 定义

就本建议书 | 国际标准而言，下列定义适用。

3.1 基本号码簿定义

下列术语在 ITU-T X.500 建议书 | ISO/IEC 9594-1 中规定：

a） 号码簿；

b） 号码簿信息库；

c） （号码簿）用户。

3.2 号码簿模型定义

下列术语在 ITU-T X.501 建议书 | ISO/IEC 9594-2 中规定：

a） 号码簿系统代理；

b） 号码簿用户代理。

3.3 号码簿信息库定义

下列术语在 ITU-T X.501 建议书 | ISO/IEC 9594-2 中规定：

a） 别名条目；

b） 号码簿信息树；

c） （号码簿）条目；

d） 直接上级；

e） 直接上级条目/对象；

f） 对象；

g） 对象类别；

h） 对象条目；

i） 下属；

j） 上级；

k） 祖先；

l） （条目的）族；

m） 复合条目。

3.4 号码簿条目定义

下列术语在 ITU-T X.501 建议书 | ISO/IEC 9594-2 中规定：

a） 属性；

b） 属性类型；

c） 属性值；

d） 属性值命题；

e） 正文；

f） 正文类型；

g） 正文值；

h） 操作属性；
i） 用户属性；
j） 匹配规则。

3.5 名称定义

以下术语在 ITU-T X.501 建议书| ISO/IEC 9594-2 中定义：

a） 别名；
b） 不同的名称；
c） （号码簿）名称；
d） 假设的名称；
e） 相对不同的名称。

3.6 分布式操作定义

以下术语在 ITU-T X.518 建议书 | ISO/IEC 9594-4 中定义：

a） 被绑定的 DSA；
b） 链接；
c） 初始执行者；
d） 被提名者。

3.7 抽象服务定义

就本建议书 | 国际标准而言，下列定义适用。

3.7.1 additional search 额外的搜索：指的是从 **joinBaseObject** 开始的一次搜索，由发起者在 **search** 请求中规定。

3.7.2 contributing member 起作用的成员：复合条目中的一个族成员，它对读、搜索或修改条目操作起作用。

3.7.3 explicitly unmarked entry 明确未标记的条目：指的是一个条目或族成员，依据管理搜索规则引用的控制属性中给出的规定，它不包括在 **SearchResult** 中。

3.7.4 family grouping 族组：复合属性的一组成员，出于操作评估目的，将它们归在一起。

3.7.5 filter 过滤器：有关条目某个属性出现或值的命题，以便限制搜索范围。

3.7.6 originator 发起者：启动一个操作的用户。

3.7.7 participation member 参与成员：指的是一个族成员，它是一个起作用的成员或是一个族组成员，总体上匹配一个 **search** 过滤器。

3.7.8 primary search 主搜索：从 **baseObject** 开始的搜索，由发起者在 **search** 请求中规定。

3.7.9 relaxation 宽松：在搜索期间对过滤器行为所做的渐进修改，以便获得更加匹配的条目，如果接收的太少，或若接收的太多而匹配的条目太少。

3.7.10 service controls 服务控制：作为操作一部分传送的参数，用于约束其性能的各个方面。

3.7.11 strand 串：包含一个通路上所有成员的族组，从叶族成员一直到祖先。族成员将驻于各串中，串的数量为其下叶族成员的数量（直接或非直接下属）。

3.7.12 streamed result 流结果：包括在多个响应 中的一个单个操作的结果。

4 缩写词

就本建议书 | 国际标准而言，下列缩写词适用。

AVA 属性值命题

DIB 号码簿信息库

DIT 号码簿信息树

DMD 号码簿管理域

DSA 号码簿系统代理

DUA 号码簿用户代理

RDN 相对不同的名称

5 惯例

除少数例外之外，本号码簿规范是依据 2001 年 11 月的 *ITU-T | ISO/IEC 公共文本描述规则*准备的。

术语“号码簿规范”（如在“本号码簿规范”中）指的是 ITU-T X.511 建议书| ISO/IEC 9594-3。术语“号码簿规范”指的是 X.500 系列建议书和 ISO/IEC 9594 的所有组成部分。

本号码簿规范使用术语*第一版本系统*来指符合号码簿规范第一版本要求的系统，即 1988 年版的 CCITT X.500 系列建议书和 1990 年版的 ISO/IEC 9594。本号码簿规范使用术语*第二版本系统*来指符合号码簿规范第二版本要求的系统，即 1993 年版的 ITU-T X.500 系列建议书和 1995 年版的 ISO/IEC 9594。本号码簿规范使用术语*第三版本系统*来指符合号码簿规范第三版本要求的系统，即 1997 年版的 ITU-T X.500 系列建议书和 1998 年版的 ISO/IEC 9594。本号码簿规范使用术语*第四版本系统*来指符合号码簿规范第四版本要求的系统，即 2001 年版的 ITU-T X.500、X.501、X.511、X.518、X.519、X.520、X.521、X.525 和 X.530 建议书，2000 年版的 ITU-T X.509 建议书，以及 2001 年版的 ISO/IEC 9594 的第 1 部分到第 10 部分。

本号码簿规范使用术语*第五版本系统*来指符合号码簿规范第五版本要求的系统，即 2005 年版的 ITU-T X.500、X.501、X.509、X.511、X.518、X.519、X.520、X.521、X.525 和 X.530 建议书，以及 2005 年版的 ISO/IEC 9594 的第 1 部分到第 10 部分。

本号码簿规范用粗体 Helvetica 字体来表示 ASN.1 记法。当在正常文本中引用 ASN.1 类型和值时，通过用粗体 Helvetica 字体表示来将它们与正常文本区分开来。典型地，在规定处理语义时所引用的程序名称与正常的文本是不同的，它们用黑体 Times 来显示。访问控制许可以斜体 Times 来表示。

如果清单中的各术语是编了号的（使用“－”或字母），那么各术语将被认为是某个程序中的各个步骤。

6 号码簿服务概述

如 ITU-T X.501 建议书 | ISO/IEC 9594-2 中所描述，号码簿服务通过 DUA 的访问点提供，每个代表一个用户开展活动。这些概念如图 1 描述。通过访问点，利用众多号码簿操作，号码簿为其用户提供服务。

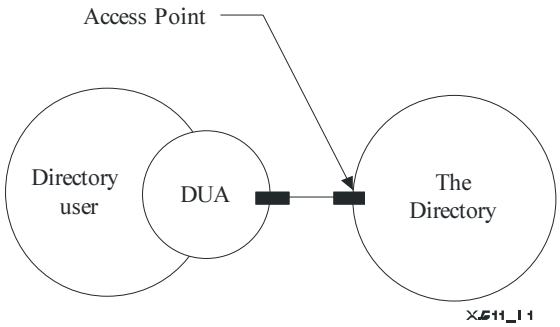

图 1－访问号码簿

有三种不同类型的号码簿操作：

a） 号码簿读操作，它查询一个单个号码簿条目；

b） 号码簿搜索操作，它潜在地查询若干号码簿条目；以及

c） 号码簿修改操作。

号码簿读操作、号码簿搜索操作和号码簿修改操作分别在第 9 节、第 10 节和第 11 节中规定。号码簿操作的一致性在 ITU-T X.519 建议书| ISO/IEC 9594-5 中规定。

7 信息类型和公共程序

7.1 引言

本节用于确定并在某些情况下用于定义众多信息类型，它们将在之后用于号码簿操作的定义中。所涉及的信息类型是那些多个操作公用的信息类型，它们有可能在今后用到，或者足够复杂或独立，值得独立于使用它们的操作单独进行定义。

号码簿服务定义中使用的若干信息类型确实在其他地方进行了定义。第 7.2 节确定了这些类型，并指明了其定义的来源。第 7.3 节到第 7.10 节的每一节都确定和定义了一种信息类型。

本节还规定了若干程序公用元素，它们适用于大多数或所有的号码簿操作。

7.2 在其他地方定义的信息类型

以下信息类型在 ITU-T X.501 建议书 | ISO/IEC 9594-2 中规定：

a） **Attribute**；

b） **AttributeType**；

c） **AttributeValue**；

d） **AttributeValueAssertion**；

e） **Context**；

f） **ContextAssertion**；

g） **DistinguishedName**；

h） **Name**；

i） **OPTIONALLY-PROTECTED**；

j） **OPTIONALLY-PROTECTED-SEQ**；

k） **RelativeDistinguishedName**。

以下信息类型在 ITU-T X.520 建议书 | ISO/IEC 9594-6 中规定：

a） **PresentationAddress**。

以下信息类型在 ITU-T X.509 建议书| ISO/IEC 9594-8 中规定：

a） **Certificate**；

b） **SIGNED**；

c） **CertificationPath**。

以下信息类型在 ITU-T X.880 建议书 | ISO/IEC 13712-1 中规定：

a） **InvokeId**。

以下信息类型在 ITU-T X.518 建议书 | ISO/IEC 9594-4 中规定：

a） **OperationProgress**；

b） **ContinuationReference**。

7.3 公共变量

CommonArguments 信息可以用来限制对号码簿所能执行各操作的调用。

```
CommonArguments ::= SET {
        serviceControls         [30]    ServiceControls DEFAULT { },
        securityParameters      [29]    SecurityParameters OPTIONAL,
        requestor               [28]    DistinguishedName OPTIONAL,
        operationProgress       [27]    OperationProgress
                                        DEFAULT { nameResolutionPhase notStarted },
        aliasedRDNs             [26]    INTEGER OPTIONAL,
        criticalExtensions      [25]    BIT STRING OPTIONAL,
        referenceType           [24]    ReferenceType OPTIONAL,
        entryOnly               [23]    BOOLEAN DEFAULT TRUE,
        nameResolveOnMaster     [21]    BOOLEAN DEFAULT FALSE,
        operationContexts       [20]    ContextSelection OPTIONAL,
        familyGrouping          [19]    FamilyGrouping DEFAULT entryOnly }
```

ServiceControls 分量在第 7.5 节中规定。它不出现将被认为相当于控制集为空。

SecurityParameters 分量在第 7.10 节中规定。如果由请求方标记操作变量，那么 **SecurityParameters** 分量应包括变量中。**SecurityParameters** 分量不出现将被认为相当于一个空集。

requestor 不同的名称确定某个特定操作的发起者。它持有在绑定号码簿之时确定的用户名称。在标记请求时可能需要它（见第 7.10 节），并将持有发起请求的用户的名称。

注 1 — 当用户拥有一个由正文区分的、可选的不同名称时，用作 **requestor** 值的名称将是所知的、主要的不同名称。否则，基于 **requestor** 值的认证和访问控制可能无法按要求开展工作。

OperationProgress、**referenceType**、**entryOnly**、**exclusions** 和 **nameResolveOnMaster** 分量在 ITU-T X.518 建议书| ISO/IEC 9594-4 中定义。它们在以下情况下由 DUA 提供：

a） 当依据继续引用工作时，它由 DSA 返回，用于响应之前的操作，其值由 DUA 复制自继续引用；或

b） 当 DUA 代表一个管理用户时，它管理 DSA 信息树，**manageDSAIT** 选项在服务控制中设置。

aliasedRDNs 分量向 DSA 指明，操作的 **object** 分量通过废弃之前操作尝试的别名来创建。整数值指明名称中的 RDN 数量，它来自别名废弃。（值将在之前操作的提名响应中设置。）

注 2 — 提供本分量的目的是为了实现与号码簿第一版本实现方案的兼容。依据号码簿规范之后版本实现的 DUA（和 DSA）将总是从后续请求的 **CommonArguments** 中省略该参数。这样，如果废弃更多的别名，那么号码簿将不发出错误信号。

operationContexts 分量提供了一系列正文命题，它们适用于在本操作中生成的属性值命题和条目信息选择，否则它们不包含同一属性类型和正文类型的正文命题。如果 **operationContexts** 不出现，或不描述某个特定的属性类型或正文类型，那么 DSA 将使用缺省的正文命题，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 7.6.1 节、第 8.9.2.2 节和第 12.8 节所述。如果选择了 **allContexts**，那么所有属性类型的所有正文都将是有效的，DSA 提供的各正文缺省值都将被覆盖（**ContextSelection** 在第 7.6 节中定义）。

familyGrouping 用于描述应选择哪个族成员，以供某个给定操作处理。在第 7.3.2 节中对它有更为详细的描述。

7.3.1 重要扩展

criticalExtensions 分量提供了一种机制，用于列出一系列对号码簿操作性能至关重要的扩展。如果扩展操作的发起者希望指明操作将以一个或多个扩展进行执行（即没有这些扩展执行操作是不可接受的），这通过设置对应扩展的 **criticalExtensions** 为来实现。如果号码簿或其一部分不能执行重要扩展，那么它返回一个 **unavailableCriticalExtension** 指示（作为 **serviceError** 或 **PartialOutcomeQualifier**）。如果号码簿不能执行不是重要的扩展，那么它不管扩展是否出现。

本号码簿规范不建立有关执行 DSA 对其所接收的 PDU 进行解码和处理的次序的规则。收到一个未知重要扩展的 DSA 将返回一个带问题 **unavailableCriticalExtension** 的 **ServiceError**，以发出信号通知操作失败。

这些号码簿规范定义了众多扩展。各扩展采用以下形式，即比特串中的额外编号位，或者集合或序列中的额外分量，第一版本系统忽视之。对每个这种扩展，分配一个整数标识符，它是可以在 **criticalExtensions** 中设置的位号。如果扩展的重要性设为重要，那么 DUA 将在 **criticalExtensions** 中设置相应的位。如果扩展的重要性设为不重要，那么 DUA 可以或不可以设置 **criticalExtensions** 的相应位。

扩展、其标识符、许可它们的操作、建议的重要性、定义它们的章节、相应的 LDAP 控制（如果有的话），如表 1 所示。

表 1—扩展

扩　　展	标识符	操　　作	关键性	定义（小节）	LDAP控制
subentries	1	所有	非关键	7.5	1.3.6.1.4.1.4203.1.10.1
copyShallDo	2	读、比较、列表、搜索	非关键	7.5	
attribute size limit	3	读、搜索	非关键	7.5	
extraAttributes	4	读、搜索	非关键	7.6	
modifyRightsRequest	5	读	非关键	9.1	
pagedResultsRequest	6	列表、搜索	非关键	10.1	1.2.840.113556.1.4.319
matchedValuesOnly	7	搜索	非关键	10.2	1.2.826.0.1.3344810.2.3
extendedFilter	8	搜索	非关键	10.2	
targetSystem	9	增加条目	关键	11.1	
useAliasOnUpdate	10	增加条目、移去条目、修改条目	关键	11.1	
newSuperior	11	修改 DN	关键	11.4	
manageDSAIT	12	所有	关键	7.5、7.13	2.16.840.1.113730.3.4.2
useContexts	13	读、比较、列表、搜索、增加条目、修改条目、修改 DN	非关键	7.6、7.8	
partialNameResolution	14	读、搜索	非关键	7.5	
overspecFilter	15	搜索	非关键	10.1.3 f)	
selectionOnModify	16	修改条目	非关键	11.3.2	
安全性参数 — 响应	17	所有	非关键	7.10	
安全性参数 — 操作代码	18	所有	非关键	7.10	

表 1—扩展

扩　　展	标识符	操　　作	关键性	定义 （小节）	LDAP控制
安全性参数 — 属性认证通路	19	所有	非关键	7.10	
安全性参数 — 错误保护	20	所有	非关键	7.10	
SPKM 证书	21	号码簿绑定	（注 3）	8.1.1	
绑定标记 — 响应	22	号码簿绑定	非关键	8.1.1	
绑定标记 — 绑定内部算法、绑定内部密钥、配置算法和配置密钥信息	23	号码簿绑定	非关键	8.1.1	
绑定标记 — DIRQOP （废弃的）	24	号码簿绑定	非关键	8.1.1	
服务主管部门	25	读、搜索、 修改条目	关键	10.2.2、13、 ITU-T X.501 建议书| ISO/IEC 9594-2 第 16 节	
entryCount	26	搜索	非关键	10.1.3	
hierarchySelection	27	搜索	非关键	7.5	
relaxation	28	搜索	非关键	7.8	
familyGrouping	29	比较、搜索、移去条目	非关键 非关键 关键	7.3.2、7.8.3、 9.2.2、10.2、11.2.2	
familyReturn	30	读、搜索、修改条目	非关键 非关键 非关键	7.6.4、7.7.1、 9.1.3、10.2.3、 11.3.3	
dnAttributes	31	搜索	非关键	10.2.2	
友好属性	32	读、搜索	非关键	7.6、7.8.2	
放弃分页结果	33	列表、搜索	关键	7.9	
DSP 上的分页结果	34	列表、搜索	非关键	7.9	
replaceValues	35	修改条目	关键	11.3.1、11.3.2	

注 1 — 为第一扩展提供了标识符 1，对应 BIT STRING 的第 1 位。BIT STRING 的第 0 位没有使用。

注 2 — 对增加条目、移去条目、修改条目、修改 DN 使用加密的或标记的和加密的安全转换或者对任何错误或结果使用保护，要求第 2 版或更高版本的协议。

注 3 — SPKM 证书扩展至关重要，除非用在利用第 2 版或更高版本建立的关联中。

7.3.2　族组

族组允许将复合条目的单个族成员、若干个族成员或所有的族成员结合在一起，以便在操作评估之间做综合考虑。而后这些语义可以用于以下操作（如下列描述所述）：比较（定义比较属性可能处于的范围）、搜索（定义可能进行过滤的组）、移去条目（定义移去组）。下列 ASN.1 用于选择族成员。

```
FamilyGrouping ::= ENUMERATED {
        entryOnly           (1),
        compoundEntry       (2),
        strands             (3),
        multiStrand         (4) }
```

entryOnly 意味着：将在组中对操作选择的特定族成员进行考虑。这是缺省值，确保向后兼容于号码簿规范的先前版本。

compoundEntry 意味着：将把操作选择的、完整的复合条目看作是一个结合了所有属性的单元。对移去条目操作，只有当规定的对象名称是复合条目祖先的对象名称时才适用，它将引起所有的族成员被同一操作移去（依据访问控制）。

strands 意味着：操作将选择所有与族成员相关的串。该选项对移去条目操作是无效的。对搜索操作，认为单个串是用于过滤器目的。如果一个或多个串的联合属性集匹配于过滤器，那么认为复合条目匹配于过滤器。如果基对象是一个子成员，那么只考虑那些通过基对象的串。对比较操作，比较中将用到来自条目所属的所有串中所有族成员的所有属性。

multiStrand 只适用于搜索操作，用于限定族信息过滤的匹配规则。其他操作忽略之。它规定每次只考虑来自复合条目中每个组的一个串，但所有结合在一起考虑。如果基对象是一个子族成员，那么 **multiStrand** 不适用，在这种情况下，**multiStrand** 将被忽略，**entryOnly** 将被替换。

7.4 公共结果

CommonResults 或 **CommonResultsSeq** 信息用于限制号码簿能执行的各检索操作的结果。另外，它出现在任何返回的错误中。

```
CommonResults ::= SET {
    securityParameters      [30]    SecurityParameters                      OPTIONAL,
    performer               [29]    DistinguishedName                       OPTIONAL,
    aliasDereferenced       [28]    BOOLEAN                                 DEFAULT FALSE,
    notification            [27]    SEQUENCE SIZE (1..MAX) OF Attribute     OPTIONAL }

CommonResultsSeq ::= SEQUENCE {
    securityParameters      [30]    SecurityParameters                      OPTIONAL,
    performer               [29]    DistinguishedName                       OPTIONAL,
    aliasDereferenced       [28]    BOOLEAN                                 DEFAULT FALSE,
    notification            [27]    SEQUENCE SIZE (1..MAX) OF Attribute     OPTIONAL }
```

注 — **CommonResults** 和 **CommonResultsSeq** 由相同的分量组成。当由类型的 **COMPONENT** 包括在集合类型中时，使用前者，同样地，后者用在序列类型中。

SecurityParameters 分量在第 7.10 节中规定。如果号码簿对结果进行标记，那么 **SecurityParameters** 分量将包括在结果中。**SecurityParameters** 分量不出现将被认为相当于一个空集。

performer 不同的名称用于确定某个特定操作的执行者。当对结果进行标记时可能需要它（见第 7.10 节），并将持有标记结果的 DSA 的名称。

当作为操作目标的对象或基对象的假设名称包括任何已废弃的别名时，**aliasDereferenced** 分量将被设为 **TRUE**。

notification 分量将用于限定返回结果和错误 APDU，例如用于提供更加精确的错误信息。标准的通告属性在 ITU-T X.520 建议书 | ISO/IEC 9594-6 的第 5.12 节中定义。此类通告属性不必储存在号码簿条目中。

7.5 服务控制

ServiceControls 参数包含控制，如果有的话，用于指导或约束服务的提供。

```
ServiceControls ::= SET {
    options                 [0] ServiceControlOptions DEFAULT { },
    priority                [1] INTEGER { low (0), medium (1), high (2) } DEFAULT medium,
    timeLimit               [2] INTEGER OPTIONAL,
    sizeLimit               [3] INTEGER OPTIONAL,
    scopeOfReferral         [4] INTEGER { dmd(0), country(1) } OPTIONAL,
    attributeSizeLimit      [5] INTEGER OPTIONAL,
    manageDSAITPlaneRef     [6] SEQUENCE {
        dsaName                     Name,
```

```
        agreementID                 AgreementID } OPTIONAL,
    serviceType                 [7] OBJECT IDENTIFIER OPTIONAL,
    userClass                   [8] INTEGER OPTIONAL }

ServiceControlOptions  ::=  BIT STRING {
        preferChaining              (0),
        chainingProhibited          (1),
        localScope                  (2),
        dontUseCopy                 (3),
        dontDereferenceAliases      (4),
        subentries                  (5),
        copyShallDo                 (6),
        partialNameResolution       (7),
        manageDSAIT                 (8),
        noSubtypeMatch              (9),
        noSubtypeSelection          (10),
        countFamily                 (11),
        dontSelectFriends           (12),
        dontMatchFriends            (13),
        allowWriteableCopy          (14) }
```

options 分量包含众多指示，若设置，则每个指示用于提出建议的条件。因此：

a） **preferChaining** 指明，优先选择的是将链接而不是提名提供给服务。不强迫号码簿依从该优先选择。

b） **chainingProhibited** 指明，禁止链接以及其他有关号码簿的请求分发方法。

c） **localScope** 指明操作限于局部范围。该选项的定义本身是一个局部问题，例如，在一个单个 DSA 或一个单个 DMD 内。

d） **dontUseCopy** 指明，拷贝的信息（如 ITU-T X.518 建议书| ISO/IEC 9594-4 中定义）不会用于提供服务。

e） **dontDereferenceAliases** 指明，不会废弃任何用于确定受操作影响的条目的别名。

注 1 — 允许引用别名条目本身是必要的，而不是被别名条目，例如为了读别名条目。

f） **subentries** 指明，搜索或列表操作仅用于访问分条目；正常的条目变得不可访问，即号码簿的行为表现仿佛显示正常的条目不存在。如果不设置该服务控制，那么操作只访问正常的条目，分条目变得不可访问。对搜索或列表之外的各操作忽略服务控制。

注 2 — 继续观察子条目对访问控制、方案、联合属性的影响，即使子条目是不可访问的。

注 3 — 如果设定该服务控制，那么可以继续将正常的条目规定为操作的基对象。

g） **copyShallDo** 指明，如果号码簿能够部分地而不是全部地满足对条目拷贝的查询要求，那么它将不链接查询。只有当不设置 **dontUseCopy** 时它才有意义。如果不设置 **copyShallDo**，那么只有当它完整得足以允许操作彻底满足拷贝要求时，号码簿才使用影像数据。由于在影像拷贝中丢失某些请求的属性，一个查询可能只能部分地满足要求，由于 DSA 不持有它没有的属性值的所有正文信息，或者由于持有影像数据的 DSA 不支持有关该数据的请求匹配规则，在影像拷贝中会丢失给定属性的某些属性值。如果设置了 **copyShallDo**，并且号码簿无法彻底满足一个查询的要求，那么它将在返回的条目信息中设置 **incompleteEntry**。

h） **partialNameResolution** 指明，如果号码簿只能解析读或搜索操作中的部分假设名称，即它将返回一个 **nameError**，那么名称包括所有已解析 RDN 的条目将被认为是操作的目标，并且在结果中将 **partialName** 设为 **TRUE**。对读或搜索之外的各操作忽略该服务控制。

注 4 — 如果设定该服务控制，那么假设的名称将是一个正文前缀条目，拒绝对其进行访问，请求方需要访问上级条目，而后将存在正文前缀条目这一情况间接地透露给请求方，即使拒绝条目的 *DiscloseOnError* 许可。

i） **manageDSAIT** 指明，管理用户已请求操作，因此对 DSA 信息树进行管理。如果在 DSA 有多个复制平面需要管理，并且 **manageDSAITPlaneRef** 服务控制未包括在操作中，那么 DSA 为操作选择一个合适的复制平面。

j） **noSubtypeMatch** 指明，不会尝试进行属性子类型匹配。除了比较和搜索操作，对其他操作将忽略该服务控制。

k） **noSubtypeSelection** 指明，不进行子类型选择。

l） **countFamily** 指明，将把复合条目的每个成员当作一个单独的条目，例如出于大小和管理限制以及宽松控制目的。如果未设置该控制，那么将把复合属性的成员当作一个单个条目。

m） **dontSelectFriends** 指明，条目信息选择中锚属性的规定不自动包括选择中的友好属性。

n） **dontMatchFriends** 指明，过滤器项中锚属性的规定只能满足锚属性值的要求，不能满足友好属性的要求。

o） **allowWriteableCopy** 指明，在提供查询服务请求中，类型 **writeableCopy** 的 DSE 是可接受的。

注 5 — **allowWriteableCopy** 服务控制不同于 **copyShallDo**，该服务控制用于指明需要一个完整的拷贝，但它不必是主要的主机，而 **copyShallDo** 用于指明任何拷贝，不论是完整的还是不完整的，都可接受。

如果忽略该分量，那么假设以下内容：对链接没有优选权，但不禁止链接；对操作范围没有限制；许可使用拷贝；将废弃别名（除非对修改操作，对它不支持别名废弃）；分条目不可访问；对不能完全满足影像数据要求的操作需做进一步链接。不过，对这些缺省，在服务特定管理区域内可以通过搜索规则进行重写。

以 **priority**（**low**、**medium** 或 **high**）优先级提供服务。注意，在号码薄中这不是一个保证的服务，整体上，不进行排队。在各基本层上使用优先级并不暗指任何关系。

timeLimit 指明服务提供中的最大耗费时间，以秒计。如果约束条件无法满足，那么报告一个错误。如果忽略该分量，那么不暗指任何时间限制。当在列表或搜索中时间限制超出时，结果是任选一个积累的结果。

注 6 — 该分量不显示流逝时间中的请求处理时间长度：在处理流逝时间中的请求时可能涉及任何数量的 DSA。

sizeLimit 仅适用于列表和搜索操作。它指明当不返回分页结果时的最大返回条目数。在超出了大小限制的情况下，列表或搜索操作的结果可以是任选一个积累的结果，数量上等于大小限制。将抛弃任何更多的结果。当返回分页结果时，执行分页的 DSA 将忽略 sizeLimit 的值，详见第 7.9 节。

scopeOfReferral 指明 DSA 返回之提名将相关的范围。依据选择的值是 **dmd** 还是 **country**，将只返回选定范围内的其他 DSA 提名。这适用于 **referral** 错误以及 **list** 和 **search** 结果 **unexplored** 参数中的提名。

attributeSizeLimit 指明任何属性的最大大小（即类型及其所有值），它包括在返回的条目信息中。如果一个属性超出了该限制，那么从返回的条目信息中删去其所有值，并在返回的条目信息中设置 **incompleteEntry**。采用的属性大小为其在持有数据的局部具体语法中的大小，以八位字节计。由于所用的不同数据保存方法，限制是不精确的。如果未规定该参数，那么不暗指任何限制。

注 7 — 作为条目不同名称一部分返回的属性值不受该限制所限。

priority、**timeLimit** 和 **sizeLimit** 的某些结合可能产生冲突。例如，短时间限制可能与低优先级产生冲突；高大小限制可能与低时间限制产生冲突；等等。

manageDSAITPlaneRef 指明，管理用户已请求操作，因此对 DSA 信息树的某个特定复制平面进行管理。如果未设置 **manageDSAIT** 选项，那么忽略 **manageDSAITPlaneRef** 服务控制。平面由 **dsaName** 分量（它是提供 DSA 的名称）和 **agreementID** 分量（它包含影像协议标识符）确定。

serviceType 服务控制只与 **search** 请求相关，它在一个服务特定管理区域内开始其最初的评估阶段；否则将忽略之。如果提供，那么它增加获得有用通告信息的可能性，当错误表达 **search** 请求时返回通告信息。

userClass 服务控制只与 **search** 请求相关，它在一个服务特定管理区域内开始其最初的评估阶段；否则将忽略之。它确定一个用户类别。它允许请求方规定另一个用户类别，否则将应用号码簿。如果提供，那么它还会增加获得有用通告信息的可能性，当错误表达 search 请求时返回通告信息。

7.6 条目信息选择

EntryInformationSelection 参数用于指明，在检索服务中，需要从条目请求什么信息。

```
EntryInformationSelection ::= SET {
        attributes                      CHOICE {
            allUserAttributes           [0]     NULL,
            select                      [1]     SET OF AttributeType
            -- empty set implies no attributes are requested -- } DEFAULT allUserAttributes : NULL,
        infoTypes                       [2]     INTEGER {
            attributeTypesOnly                  (0),
            attributeTypesAndValues             (1) } DEFAULT attributeTypesAndValues,
        extraAttributes                 CHOICE {
            allOperationalAttributes    [3]     NULL,
            select                      [4]     SET SIZE (1..MAX) OF AttributeType } OPTIONAL,
        contextSelection                ContextSelection OPTIONAL,
        returnContexts                  BOOLEAN DEFAULT FALSE,
        familyReturn                    FamilyReturn  DEFAULT
                                        { memberSelect  contributingEntriesOnly } }

ContextSelection ::= CHOICE {
        allContexts             NULL,
        selectedContexts        SET SIZE (1..MAX) OF TypeAndContextAssertion }

TypeAndContextAssertion ::= SEQUENCE {
        type                    AttributeType,
        contextAssertions       CHOICE {
            preference                  SEQUENCE OF ContextAssertion,
            all                         SET OF ContextAssertion } }

FamilyReturn ::= SEQUENCE {
        memberSelect   ENUMERATED {
            contributingEntriesOnly     (1),
            participatingEntriesOnly    (2),
            compoundEntry               (3) },
        familySelect  SEQUENCE SIZE (1..MAX) OF OBJECT-CLASS.&id   OPTIONAL }
```

attributes 分量用于规定有关请求信息的用户和操作属性。

a） 如果选择 **select** 选项，那么列出涉及的各属性。如果清单为空，那么将不返回任何属性。如果属性出现，那么将返回有关所选属性的信息。如果所选的属性一个也没有出现，那么将只返回带问题 **noSuchAttributeOrValue** 的 **attributeError**。

b） 如果选择 **allUserAttributes** 选项，那么请求有关条目中所有用户属性的信息。

只有当有足够的访问权限时，才返回属性信息。只有在访问权限排除读所有请求的属性值的情况下，才返回 **securityError**（带问题 **insufficientAccessRights**）。注意，访问控制还适用于依据 **EntryInformationSelection** 分量适于返回的各属性和值，并可进一步减少返回的信息。

注 1 — 访问控制还适用于依据 **EntryInformationSelection** 分量适于返回的各属性和值，并可进一步减少返回的信息。

infoTypes 分量用于规定是属性类型和属性值信息（缺省）都请求，还是只请求属性类型信息。如果一个类型的某个属性是其他属性的载体，例如，**family-information** 属性，那么将独立于 **infoTypes** 分量的设置返回值，当 **infoTypes** 规定将适用于所包含的属性。如果 **attributes** 分量不请求任何属性，那么该分量没有意义。

extraAttributes 分量用于规定额外的用户集，以及有关请求信息的操作属性。如果选择了 **allOperationalAttributes** 选项，那么请求的信息有关条目中的所有号码薄操作属性。如果选择了 **select** 选项，那么请求的信息有关列出之属性。

注 2 — 该分量可以用于请求以下信息，例如，当 **attributes** 设为 **allUserAttributes** 时的特定操作属性，或者所有的操作属性。如果在 **attributes** 和 **extraAttributes** 中都列出或暗指了相同的属性，那么作为只请求了一次对待。

如果未设置 **noSubtypeSelection** 服务控制选项，那么有关某个特殊属性的请求总被看作是对该属性及其所有子类型的请求（除了由第一版本系统处理的请求）。如果设置了 **noSubtypeSelection** 服务控制选项，那么只返回请求的属性，而不返回其子类型。同样，如果 **dontSelectFriends** 服务控制选项没有设置，那么有关某个拥有友好属性的特殊属性的请求，总被看作是对该属性及其所有友好属性的请求。

在响应一个有关属性信息的请求时，号码薄在对待条目的所有联合属性时就当它们仿佛是条目的实际用户属性，即像其他用户属性一样来选择它们，并合并进返回的条目信息中。有关 **allUserAttributes** 的请求将请求条目的所有联合属性，以及条目的普通属性。如果以下所有各项都为 TRUE，那么属性是条目的一个联合属性：

a） 它位于子条目中，其子树规范包括条目；

b） 它可以出现在等同于联合属性类型的 **collectiveExclusions** 属性值条目中；以及

c） 内容规则认可条目的结构对象类别。

contextSelection 分量用于规定返回 **attributes** 或 **extraAttributes** 所选之属性中的哪个属性值。只对以下属性值的 **contextSelection** 进行评估，即依据 **EntryInformationSelection** 的其他分量，它们是候选的返回属性。如果它不提供，那么 **contextSelection** 的评估、缺省值的使用将在第 7.6.1 节到第 7.6.3 节论述。

如果 **infoTypes** 分量不请求任何属性值，或者 **attributes** 分量不请求任何属性，那么 **contextSelection** 分量没有意义。如果作为应用 **contextSelection** 的结果，没有任何属性值适于返回，那么可以不带任何值地返回属性。

returnContexts 分量用于请求号码簿带其相关正文清单地返回属性值。如果该分量不存在或用一个 **FALSE** 值来规定，那么在结果中将不返回任何正文信息。如果该分量用一个 **TRUE** 值来规定，那么对每个返回的属性值，返回所有的正文信息。注意，当 **returnContexts** 为 **TRUE** 时，**contextSelection** 分量对返回什么正文信息不产生选择性的影响。

如果已标记了一个或多个族成员，那么 **familyReturn** 分量（如果存在的话）用于确定复合条目中的哪些条目将被返回（见第 7.6.4 节）。

7.6.1 使用contextSelection或正文选择缺省值

contextSelection 分量用于选择 **attributes** 或 **extraAttributes** 所选属性的某些属性值。只能依据候选返回属性值对 **contextSelection** 进行评估，候选属性值依据 **EntryInformationSelection** 的其他分量返回。对每个属性值，管理其属性的任何内容选择都应评估为 TRUE（在第 7.6.2 节中定义），以便选择该属性值。

如果出现任何下列条件，那么 **contextSelection** 用于管理属性类型：

— **ContextSelection** 用于规定 **allContexts**（在这种情况下，选择所有属性类型的所有属性值）；

— **ContextSelection** 拥有一个 **selectedContexts**，它包括一个 **TypeAndContextAssertion**，其类型等同于属性类型或属性类型的父类型；或者

— **ContextSelection** 拥有一个 **selectedContexts**，它包括一个 **TypeAndContextAssertion**，其类型为 **id-oa-allAttributeTypes**。

如果不提供 **contextSelection**，或它不管理给定的属性类型，那么将应用一个缺省的 **contextSelection**。除了 **EntryInformationSelection** 中的 **contextSelection**，还有三个潜在的 **contextSelection** 来源：整体上为操作规定的 **contextSelection**；在 DIT 各分条目中可用的 **contextSelection**；在 DSA 中局部可用的 **contextSelection**。依据以下优先权来使用它们：

1） 如果 **contextSelection** 出现在 **EntryInformationSelection** 中，并且它管理给定的属性类型，如上所述，那么将应用它。

2） 如果 **contextSelection** 未出现在 **EntryInformationSelection** 中，或者它出现了但不管理给定的属性类型，那么如果一个出现并且管理给定的属性类型，如上所述，那么将应用如第 7.3 节所述已经为操作提供的 **operationContexts**。

3） 如果请求既不是 **EntryInformationSelection** 中的 **contextSelection**，也不是操作的 **operationContexts**，或者也不管理给定的属性，那么将应用在控制条目的各正文命题分条

目（如果有的话）中的 **contextAssertionDefaults** 属性值，作为 **selectedContexts**（正文命题分条目在 ITU-T X.501 建议书| ISO/IEC 9594-2 第 14.7 节描述）。

4） 如果上面描述的来源中没有 **contextSelection** 来管理给定的属性类型，那么 DSA 可以应用一个局部定义的缺省 **contextSelection**。这样一个缺省值将典型地反映局部参数，例如语言、DSA 的部署位置、当前时间，但对其响应的每个 DUA，可以由 DSA 做不同的剪裁。

5） 如果任何这些来源中都没有任何 **contextSelection** 来管理给定的属性类型，那么认为选择了该属性的所有值（即将 **allContexts** 当作基缺省值）。

注 — 除了管理相同属性类型但发表一个有关不同正文类型的命题的早期 **contextSelection** 外，还将应用管理给定属性类型并发表一个有关某个正文类型的命题的缺省 **contextSelection**，优先级顺序同上所述。

7.6.2 评估contextSelection

contextSelection 将为 TRUE（即选择一个给定的属性值），若：

a） 规定了 **allContexts**（这允许一个正文选择覆盖任何缺省值，否则如果省略该 **contextSelection**，那么应用缺省值。）；或者

b） **selectedContexts** 中的每个 **TypeAndContextAssertion** 都为 TRUE，如第 7.6.3 节所述。

否则 **contextSelection** 为 FALSE。

7.6.3 评估TypeAndContextAssertion

TypeAndContextAssertion 将为 TRUE（即选择一个给定的属性值），若：

a） 属性类型不同于 **TypeAndContextAssertion** 中的 **type**（也不是其子类型），**TypeAndContextAssertion** 中的 **type** 不是 **id-oa-allAttributeTypes**。在这种情况下，**TypeAndContextAssertion** 不适用于特定属性值的属性类型，因此不从选择中去除属性值；或者

b） 对属性值，**TypeAndContextAssertion** 中的 **contextAssertions** 为 TRUE，如下定义。

注 1 — **OBJECT IDENTIFIER** 的值 **id-oa-allAttributeTypes** 可以用作 **TypeAndContextAssertion** 中的 **type** 值，以便推动依据属性类型的属性值对 **contextAssertions** 进行评估。

contextAssertions 表示为一个有关首选正文的有序序列，或一个有关正文命题的复合集：

a） 如果规定为 **all**，那么只有当 SET 中的每个 **ContextAssertion** 都为 TRUE 时，**contextAssertions** 才为 TRUE，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 8.9.2.4 节所定义。

b） 如果规定为 **preference**，那么依据相同属性类型的所有候选属性值依次对 **SEQUENCE** 中的每个 **ContextAssertion** 进行评估，直至 **ContextAssertion** 评估为 TRUE，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 8.9.2.4 节所定义。（**fallback** 标志如果出现，在整个 SEQUENCE 耗尽之前不对它作考虑。）一旦对其中之一的候选属性值 **ContextAssertion** 评估为 TRUE，那么将对相同属性类型的每个候选属性值都进行评估，但忽略 SEQUENCE 中的后续 **ContextAssertion**。

注 2 — **preference** 提供了一种选择方式，以正文第一选择、第二选择等形式进行规定（例如，语言＝法语，若没有法语，则语言＝英语）。

否则 **TypeAndContextAssertion** 为 FALSE。

7.6.4 族返回

如果一个或多个族成员已标记为起作用的成员或参与成员，那么 **familyReturn** 分量用于确定将返回复合条目中的哪些条目。有关如何标记族成员的程序在第 7.13 节中做进一步描述。

memberSelect 分量规定在结果中选择哪些条目予以返回：

— **contributingEntriesOnly** 意味着只返回由操作标记为起作用成员的族成员。在读或修改条目操作的情况下，它为由 **object** 操作变量确定的族成员，对搜索操作，它包括对匹配有影响的族成员。

— **participatingEntriesOnly** 意味着只返回由操作标记为参与成员的族成员。在读或修改条目操作的情况下，同 **contributingEntriesOnly**。

— **compoundEntry** 意味着将返回复合条目中的每个族成员，除了那些在搜索操作中可能由管理搜索规则明确未标记的成员。

除了 **memberSelect** 规定的内容，通过规定返回选定族的所有下属成员，**familySelect** 分量对 **memberSelect** 分量进行补充。元素的序列并不重要。族由祖先的族成员直接下属的结构对象类别确定。如果 **memberSelect** 规定了 **compoundEntry**，那么该分量没有作用。

注 — 一个管理搜索规则可以修改应返回什么信息（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 16.10 节）。

7.7 条目信息

7.7.1 条目信息数据类型

EntryInformation 数据类型从条目传送所选的信息。

```
EntryInformation ::= SEQUENCE {
        name                            Name,
        fromEntry                       BOOLEAN DEFAULT TRUE,
        information                     SET SIZE (1..MAX) OF CHOICE {
            attributeType               AttributeType,
            attribute                   Attribute } OPTIONAL,
        incompleteEntry         [3]     BOOLEAN DEFAULT FALSE, -- 不在第一版本系统中
        partialName             [4]     BOOLEAN DEFAULT FALSE, -- 不在第一或第二版本系统中
        derivedEntry            [5]     BOOLEAN DEFAULT FALSE -- 不在第四版本之前系统中 -- }
```

Name 参数指的是条目的不同名称或者条目的别名名称。无论何时当准许访问控制策略，则返回条目的不同名称。如果允许对条目的属性而非其不同名称进行访问，那么号码簿可以返回一个错误或条目有效别名的名称。

主要的不同名称用于 **Name** 参数。这意味着如果形成名称的 RDN 包括一个具有多个不同值（由正文区分）的属性，那么主要的不同值将当作该属性返回 RDN **AttributeTypeAndDistinguishedValue** 中的值来使用。由于对每个 RDN，返回的 **value** 总为主要的不同值，因此将为所有的 **AttributeTypeAndDistinguishedValue** 删去 **primaryDistinguished**。

只有当正文选择已用于返回的条目信息时，**Name** 中的 RDN 才包括可选的不同值。可选的不同名称作为返回 RDN **AttributeTypeAndDistinguishedValue** 中 **valuesWithContext** 的一部分返回。适用于返回条目信息的正文选择（见第 7.6.1 节）也适用于可选的不同值，用于确定在 **valuesWithContext** 中使用哪个不同的值。

注 1 — 内容选择不适用于在 **Name** 中返回的、主要的不同值。

如果已经请求利用结果返回正文信息，那么正文信息还将包括在的 **Name** 中不同值可用的地方（利用 RDN 的 **valuesWithContext** 元素）。当返回可选的不同值时，总为所有的不同值返回正文信息。

注 2 — 如果使用别名定位条目，那么该别名应为一个有效的别名。否则，它如何确保别名是有效的将处于这些号码簿规范的范畴之外。

注 3 — 当号码簿的某个特定分量选择返回一个它可用的别名时，建议在以下地方进行选择，即它可能为同一请求方提出的重复请求选择相同的别名，以便提供一致的服务。

fromEntry 参数用于指明信息是取自条目（**TRUE**）还是取自条目的拷贝（**FALSE**）。

如果返回条目中的任何属性信息，那么包括 **information** 参数，合适的话，包含一系列 **attributeTypes** 和 **attributes**。

无论何时当与用户请求相关的返回条目信息不完整时，包括 **incompleteEntry** 参数，并设为 **TRUE**，例如，由于出于访问控制原因删去属性或属性值（以及允许透露其存在情况），由于出现不完整的影像信息以及 **copyShallDo**，或者由于超出了 **attributeSizeLimit**。由于已返回别名名称而不是不同名称，因此不设为 **TRUE**。

在考虑 **partialNameResolution** 服务控制之前，号码簿将在整体上完成操作的名称解析阶段（包括在提名后，检查所有的相关知识引用，等等）。如果已耗尽所有的名称解析选项，并且已至少解析一个 RDN，那么将包括 **partialName** 参数，如果请求已设置 **partialNameResolution** 服务控制，并且号码簿无法完成对相关条目所有 RDN 的名称解析，那么设为 **TRUE**。当 **partialName** 返回为 **TRUE** 时，它指明返回的信息来自条目，位置在成功解析最后一个 RDN 的地方。

无论何时当返回的条目信息包含连接结果（通过对源自多个号码簿条目的数据执行连接而获得），则包括 **derivedEntry** 参数，并设为 **TRUE**。当该参数为 **TRUE** 时，**name** 的值可以是任何相关条目（条目信息源自这些条目）的名称，或者是任何这些条目的别名名称。**name** 的值不得用在后续操作中。如果 **derivedEntry** 参数设为 **TRUE**，并且签署了响应，那么签名为执行连接的 DSA 的签名。

7.7.2 条目信息中的族信息

当返回来自复合条目的信息时，则依据 **EntryInformationSelection**（管理搜索规则可能对之进行修改）来选择每个待返回成员的属性。当在 **search** 请求中设置了 **separateFamilyMembers** 搜索控制选项时，则每个成员作为一个单独的条目返回。否则，如果返回多个成员，那么条目信息将以如下方式进行包装，即信息看起来像是来自一个单个条目，它可以是祖先或者是一个下属成员（当 **search** 请求的基对象为下属于祖先的族成员并且 **FamilyReturn** 尚未选择祖先时，后者是合适的）。其他成员的属性将包装进一个 **family-information** 派生属性中，如下所述。

注 1 — 依据上述内容，多个族成员总是包装在一个 **read** 或 **modifyEntry** 结果中。

family-information 衍生属性仅用于包装；属性不作为不同的实体存在；它不能直接由 **entryInformationSelection** 选择（将忽略任何有关这方面的尝试），也不直接受访问控制保护。

```
family-information  ATTRIBUTE  ::= {
        WITH SYNTAX             FamilyEntries
        USAGE                   directoryOperation
        ID                      id-at-family-information }

FamilyEntries  ::=  SEQUENCE  {
        family-class            OBJECT-CLASS.&id,       -- 结构对象类别值
        familyEntries           SEQUENCE OF FamilyEntry }

FamilyEntry  ::=  SEQUENCE  {
        rdn                     RelativeDistinguishedName,
        information             SEQUENCE OF CHOICE  {
            attributeType               AttributeType,
            attribute                   Attribute },
        family-info             SEQUENCE SIZE (1..MAX) OF FamilyEntries OPTIONAL }
```

family-information 属性是一个多值属性。如果祖先指定为信息源，那么每个属性值持有来自一个单个族的信息。如果作为祖先下属的一个族成员指定为信息源，那么基于指定成员直接下属成员的结构对象类别，信息归类为属性值。

选定的每个族成员通过类型 **FamilyEntry** 的一个值来描述，它包含：

— 选定的属性信息（在适当的地方），作为一个属性类型，或作为一个完整的属性，它取决于 **EntryInformationSelection** 中的 **infoTypes** 值；

注 2 — 如第 7.6 节所述，**infoTypes** 规定只适用于所含的属性，不适用于 **family-information** 属性自身。

— 任何以完整的 **family-information** 属性形式出现的、嵌套的 **FamilyEntries** 信息，依据下属条目的结构对象类别结合在一起；

— 根本不对未选条目进行描述，除非它们是一个或多个所选族成员的上级。

7.8 过滤器

7.8.1 过滤器参数

Filter 参数用于测试是否满足某个特定条目的要求。过滤器以有关条目某些属性是否出现或值的命题的形式进行表示，当且仅当评估值为 TRUE 时，它才满足要求。

注 — 一个过滤器可以是 TRUE、FALSE 或 UNDEFINED（未定义）。

```
Filter ::= CHOICE {
        item    [0]    FilterItem,
        and     [1]    SET OF Filter,
        or      [2]    SET OF Filter,
        not     [3]    Filter }

FilterItem    ::=    CHOICE {
        equality                [0]    AttributeValueAssertion,
        substrings              [1]    SEQUENCE {
            type                           ATTRIBUTE.&id ({ SupportedAttributes }),
            strings                        SEQUENCE OF CHOICE {
                initial                    [0]    ATTRIBUTE.&Type
                                                  ({SupportedAttributes}{@substrings.type}),
                any                        [1]    ATTRIBUTE.&Type
                                                  ({SupportedAttributes}{@substrings.type}),
                final                      [2]    ATTRIBUTE.&Type
                                                  ({SupportedAttributes}{@substrings.type}),
                control                           Attribute } },  -- 用于规定以下项的解释
        greaterOrEqual          [2]    AttributeValueAssertion,
        lessOrEqual             [3]    AttributeValueAssertion,
        present                 [4]    AttributeType,
        approximateMatch        [5]    AttributeValueAssertion,
        extensibleMatch         [6]    MatchingRuleAssertion,
        contextPresent          [7]    AttributeTypeAssertion }

MatchingRuleAssertion ::= SEQUENCE {
        matchingRule            [1]    SET SIZE (1..MAX) OF MATCHING-RULE.&id,
        type                    [2]    AttributeType  OPTIONAL,
        matchValue              [3]    MATCHING-RULE.&AssertionType ( CONSTRAINED BY {
                -- matchValue 应是一个类型值，由 matchingRule 确定的其中一个
                -- MATCHING-RULE 的&AssertionType 字段规定。 -- } ),
        dnAttributes            [4]    BOOLEAN DEFAULT FALSE }
```

一个 **Filter** 是一个 **FilterItem**（见第 7.8.2 节），或者是一个涉及更简单过滤器的表达式，过滤器由逻辑运算符 **and**、**or** 和 **not** 合成。过滤器的评估结果受宽松策略行为的影响，它可引起用一个匹配规则替代另一个匹配规则，或可提供认为匹配的值。

一个 **Filter** 是一个具有 **FilterItem** 的值（即 TRUE、FALSE 或 UNDEFINED）的 **FilterItem**。

如果设置为空或者如果每个过滤器都为 TRUE，那么一个为一系列过滤器 **and**（“与”）运算结果的 **Filter** 值为 TRUE；如果至少一个为 FALSE，那么它为 FALSE；否则它为 UNDEFINED（即至少一个过滤器为 UNDEFINED，并且没有任何一个过滤器为 FALSE）。

如果设置为空或者如果每个过滤器都为 FALSE，那么一个为一系列过滤器 **or**（“或”）运算结果的 **Filter** 值为 FALSE；如果至少一个为 TRUE，那么它为 TRUE；否则它为 UNDEFINED（即至少一个过滤器为 UNDEFINED，并且没有任何一个过滤器为 TRUE）。

如果过滤器为 FALSE，那么一个为某个过滤器 **not**（“非”）运算结果的 **Filter** 值为 TRUE；如果它为 TRUE，那么 **Filter** 为 TRUE；如果它为 UNDEFINED，那么 **Filter** 为 UNDEFINED。

一个非求反过滤器项定义为一个嵌入于最外 **Filter** 内偶数个（可能为 0 个）**not** 元素中的过滤器项。因此，一个只含有 **and** 或 **or** 组合中过滤器项的过滤器将只含有非求反项。一个求反过滤器项定义为一个嵌入于最外 **Filter** 内奇数个 **not** 元素中的过滤器项。

7.8.2 过滤器项

FilterItem 是一个有关所测试条目是否出现或者属性值的命题。如果条目包含属性的一个子类型，并且对子类型命题为 TRUE，并且未设置 **noSubtypeMatch** 服务控制选项，或者如果有一个联合条目属性

（见 7.6），对其命题为 TRUE，或者如果以下情况，那么有关某个特殊属性类型的命题也是满足要求的。

— 不设置 **dontMatchFriends** 服务控制选项；以及

— 条目包含一个有关规定属性的友好属性，它有一个兼容于命题的匹配规则；以及

— 对友好属性，命题为 TRUE。

每个命题为 TRUE、FALSE 或 UNDEFINED。

每个 **FilterItem** 包括或暗指一个或多个用于确定所考虑之特定属性的 **AttributeTypes**。

有关此类属性值的任何命题只有当评估机制了解 **AttributeType** 时才定义，假设的 **AttributeValue** 符合为该属性类型所定义的属性语法要求，隐含的或指明的匹配规则适用于该属性类型，并且（当使用时）出现的 **matchValue** 符合为指明的匹配规则所定义的语法要求。当不满足这些条件时，**FilterItem** 将评估为逻辑值 UNDEFINED。

注 1 — 访问控制限制可能影响对 **FilterItem** 的评估，并可能引起 **FilterItem** 被评估为 UNDEFINED。

另外，如果相关于未出现在属性（其命题正在接受测试）中的属性值和属性类型，那么由这些条件定义的命题评估为 UNDEFINED。由这些条件定义并且相关于出现之属性值的命题评估 FALSE。

利用为该属性类型定义的匹配规则来评估过滤器项中的属性值命题，合适的话，依据宽松策略的行为，替代该属性类型。依据其定义中的规定对匹配规则命题进行评估。为某个特殊语法定义的匹配规则只能用于生成有关该语法属性或该语法子类型的命题。

注 2 — 宽松策略行为可引起某个特定的匹配规则回复为 **nullMatch** 匹配规则（它总评估为 TRUE（如果非求反）或 FALSE（如果求反）－见 ITU-T X.520 建议书| ISO/IEC 9594-6 第 6.7.2 节）。

一个 **FilterItem** 可以为 UNDEFINED（如上所述）。否则，当 **FilterItem** 命题为：

a） **equality** — 当且仅当有一个属性值或者 **equality** 匹配规则的其中一个子类型应用于该值并且出现的值返回 TRUE，它才为 TRUE。

b） **substrings** — 当且仅当有一个属性值或者 **substring** 匹配规则的其中一个子类型应用于该值并且在 **strings** 中出现的值返回 TRUE，它才为 TRUE。有关出现的值的语义描述见 ITU-T X.520 建议书| ISO/IEC 9594-6。

c） **greaterOrEqual** — 当且仅当有一个属性值或者 **ordering** 匹配规则的其中一个子类型应用于该值并且出现的值返回 FALSE，即有一个属性值大于或等于出现的值，它才为 TRUE。

d） **lessOrEqual** — 当且仅当有一个属性值或者 **equality** 匹配规则或 **ordering** 匹配规则的其中一个子类型应用于该值并且出现的值返回 TRUE，即有一个属性值小于或等于出现的值，它才为 TRUE。

e） **present** — 当且仅当属性值或者其中一个子类型出现在条目中，它才为 TRUE。

f） **approximateMatch** — 当且仅当有一个属性值或者某些局部定义的近似值匹配算法（例如拼写变化、语音匹配等）的其中一个子类型返回 TRUE，它才为 TRUE。如果一个项匹配等于，那么它也将满足近似匹配。否则在本版号码簿中没有任何有关近似匹配的特定指南。如果不支持近似匹配，那么该 **FilterItem** 应当作是一个 **equality** 匹配。

g） **extensibleMatch** — 当且仅当有一个带指明 **type** 的属性值或者在 **matchingRule** 中规定了匹配规则的其中一个子类型应用于该值并且出现的值 **matchValue** 返回 TRUE，它才为 TRUE。

如果提供了若干匹配规则，那么对这些规则如何结合成一个新规则的方法不做规定（它是一个局部定义算法，反映了组成匹配规则的语义，例如 **phonetic** + **keyword** 匹配）。

如果省略 **type**，那么对所有兼容该匹配规则的属性类型进行匹配。如果 **dnAttributes** 为 **TRUE**，那么除了在评估匹配中所用的那些条目属性之外，还使用条目的不同名称属性。

如果在 **filter**（而不是在 **extendedFilter**）中请求 **extensibleMatch**，那么将对 **CommonArguments** 中 **criticalExtensions** 参数中的 **extendedFilter** 位进行设置，指明扩展是重要的。

如果实现方案不支持任何在 **matchingRule** 子分量中定义的匹配规则，或者没有任何一个匹配规则兼容属性类型，那么若未设置 **performExactly** 搜索控制选项，则 **extensibleMatch** 过滤器项评估为 UNDEFINED。如果设置了 **performExactly** 搜索控制选项，那么 **search** 请求将被以下拒绝：

— 一个带问题 **unsupportedMatchingUse** 的 **serviceError**；

— 如果不支持所有匹配规则，那么为一个带值 **id-pr-unsupportedMatchingRule** 的 **searchServiceProblem** 通告属性，否则为一个带值 **id-pr-unsupportedMatchingUse** 的 **searchServiceProblem** 通告属性；

— 一个 **attributeTypeList** 通告属性，值同定义了无效匹配规则的属性类型；以及

— 一个 **attributeTypeList** 通告属性，值同不支持与/或不兼容匹配规则的对象标识符。

注 3 — 对第一版本系统，不允许 **extensibleMatch**。

h） **contextPresent** — 当且仅当该属性类型的 **AttributeTypeAssertion** 评估为 TRUE，或者若未设置 **noSubtypeMatch** 服务控制选项，则其中一个子类型评估为 TRUE，它才为 TRUE。

如果正文命题包括在过滤器项的一个属性值命题中，那么只依据那些满足所有给定正文命题的值来对过滤器项进行评估，如 ITU-T X.501 建议书| ISO/IEC 9594-2 第 8.9.2 节所述。如果没有任何正文命题包括在属性值命题中，那么应用缺省的正文命题，如 ITU-T X.501 建议书| ISO/IEC 9594-2 第 8.9.2.2 节所述。

7.8.3 用族信息评估过滤器

在实现过滤器需求中，特定的族组工作如下：

entryOnly 意味着只有彻底实现了过滤器需求的那些族成员才标记为起作用的成员和参与成员（有关起作用的成员和参与成员的定义见第 7.13 节）。

compoundEntry 意味着整个复合条目形成组，它将满足完整过滤器的要求；在每个复合条目中，它满足过滤器的要求，对匹配起作用的族成员标记为起作用的成员，复合条目的所有成员标记均为参与条目。

strands 意味着过滤器应用于从叶到祖先的每个完整串。如果至少一个串匹配过滤器，那么复合条目匹配过滤器。对匹配起作用的匹配串上的族成员标记为起作用的成员，匹配串上的所有成员标记均为参与条目。

一个串是族内的一组成员，它们形成从叶到祖先的一条路径，由于有许多叶条目，因此会有许多串。

multiStrand 意味着对来自每个族类别的串的结合是一种出于匹配目的的族组合。所有的结合在当时都认为是一个。如果至少一个串结合匹配过滤器，那么复合条目匹配过滤器。对匹配起作用的匹配串上的族成员标记为起作用的成员，匹配串结合的所有成员标记均为参与条目。

当且仅当族成员直接下属于具有相同结构对象类别的祖先，两个串才能具有相同的*族类别*。

当且仅当它出现在至少一个可能的、引起条目匹配分过滤器的串组合中，一个串才能匹配于一个过滤器。以下是必然结果：

— 如果祖先完全匹配分过滤器，那么*所有*的串都匹配。

— 同样，如果对某个特定的祖先有三个族类别，并且两个族类别满足分过滤器要求，而不考虑第三个族类别，那么第三个族类别的所有串都匹配。

只有当基对象为 DIT 中的祖先（或更高）时，**multiStrand** 才适用。如果基对象是一个族成员，但不是一个祖先，那么将忽略 **multiStrand**，并替换 **entryOnly**。

7.9 分页的结果

DUA 利用 **PagedResultsRequest** 参数来请求将列表或搜索操作的结果“逐页地”返回给它：它请求 DSA 只返回一个子集 — 操作结果的 — 一页，特殊地，为下一个 **pageSize** 的下属或条目，并返回一个 **queryReference**，它可用于在接下来的查询中请求下一个结果集。

可以由 DSA 来执行分页结果，通过绑定操作，已将 DUA 绑定于其上（绑定的 DSA），或者可以由开始最初评估阶段的 DSA 来执行分页结果（最初的执行方如 ITU-T X.518 建议书| ISO/IEC 9594-4 第 15.5.5 节详述）。

如果将对结果进行标记，那么将不使用它，除非在 DSA 间达成合作提供分页结果的谅解，这样，执行分页的 DSA 可以从收自其他 DSA 的结果中移去签名，然后它自己对将返回给 DUA 的结果进行标记。建立这种谅解的方式在本号码簿规范的范围之外。虽然 DUA 可以请求 **pagedResults**，但允许 DSA 忽略结果，并以正常方式返回其结果。

注 1 — 在配置不是“良好连接”的情况下，结果可能是不可预测的，例如因为影像和使用 NSSR，名称解析将确定多个基对象。

如果请求分页结果并执行了分页，那么如果有的话，分页 DSA 将忽略 **sizeLimit** 服务控制。如果不执行分页，那么将重视 **sizeLimit** 服务控制。一个起作用的 DSA（见 ITU-T X.518 建议书 | ISO/IEC 9594-4 第 15.5.5 节）将重视 **sizeLimit** 服务控制。

```
PagedResultsRequest ::= CHOICE {
        newRequest                  SEQUENCE {
                pageSize                        INTEGER,
                sortKeys                        SEQUENCE SIZE (1..MAX) OF SortKey OPTIONAL,
                reverse             [1]         BOOLEAN DEFAULT FALSE,
                unmerged            [2]         BOOLEAN DEFAULT FALSE,
                pageNumber          [3]         INTEGER OPTIONAL },
        queryReference              OCTET STRING,
        abandonQuery        [0]     OCTET STRING }

SortKey ::= SEQUENCE {
        type                AttributeType,
        orderingRule        MATCHING-RULE.&id OPTIONAL }
```

对一个新的列表或搜索操作，将 **PagedResultsRequest** 设为 **newRequest**，它由以下参数组成：

a） **pageSize** 参数用于规定结果中返回的下属或条目的最大数量。DSA 返回的条目数量最多可以达到请求的条目数量，但不超过。如果有的话，将忽略 **sizeLimit**。当封装在 **family-information** 派生属性中时，是否包括族信息不取决于页的大小。

b） **sortKeys** 参数用于规定一系列属性类型，可选的次序匹配规则用作排序关键字，在返回 DUA 之前对返回的条目进行排序。在列表操作情况下，将通过 RDN 进行排序，但排序要求仅适用于 RDN 中的属性。在搜索操作情况下，排序仅适用于实际提供的属性（作为选择的结果，以不同名称排序的访问控制作为反馈）。依据序列中第一个 **SortKey** 的 **type** 属性值对各条目进行排序，在多个条目具有相同排序位置的情况下，依据序列中下一个 **SortKey** 的 **type** 属性值进行排序，等等。

对某个特殊的 **SortKey**，如果它出现，那么 DSA 使用 **orderingRule** 匹配规则，否则如果做了定义，那么使用属性的 **ordering** 匹配规则。如果属性类型是多值的，那么用“最小的”值；如果属性类型从返回的结果中丢失，那么将之看作“大于”所有其他匹配的值。允许一个 DSA 只支持某些排序主要序列（因此，按首字母内部排序的、持有并返回其数据的 DSA 将只能符合一个序列关键序列的要求）。如果它不支持请求的序列，那么它将使用一个缺省的排序序列。

不能分隔层次型组，但可以在序列中予以返回，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 10.3 节所规定的那样。当进行排序时，返回的层次型组的第一个条目将确定层次型组在排序结果中的位置。

注 2 — 一个层次型组可以跨越若干页。

c） 如果 **reverse** 参数为 **TRUE**，那么 DSA 将以倒序返回排序结果（即从“最大”到“最小”— 如果属性类型是多值的，那么使用“最大”值；如果属性类型从返回的结果中丢失，那么将

之看作“小于”所有其他匹配的值。）。如果 **reverse** 参数为 **FALSE**，那么 DSA 将以正序返回排序结果。如果没有规定任何 **sortKeys** 参数，那么该参数将被忽略。

d） 如果 **unmerged** 参数为 **TRUE**，并且负责分页的 DSA 收集来自众多其他 DSA 的结果，那么在返回来自下一个 DSA 的数据之前，它将返回来自某个 DSA 的所有数据（以排序次序）。如果 **unmerged** 参数为 **FALSE**，那么 DSA 将收集来自所有其他 DSA 的结果，并在返回任何内容之前对合并的数据进行排序。如果没有规定任何 **sortKeys** 参数，那么该参数将被忽略。不论 DSA 是否支持 DSP 分页结果，**unmerged** 参数的语义都是相同的。

e） 如果 **pageNumber** 参数出现，那么它指明用户想从某个特殊的页开始，而不必从第一页开始。如果未请求排序，那么该参数将被忽略。

对紧接着的请求，即请求分页结果的下一集合，DUA 如之前一样生成列表或搜索请求，当将 **PagedResultsRequest** 设为 **queryReference**，该参数的值等同于在前一结果的 **PartialOutcomeQualifier** 中返回的值。DUA 不了解 **queryReference**，它供 DSA 使用，原因是它希望为该查询记录正文信息。DSA 使用该信息来确定下一个要返回的结果。

通过生成如之前一样的 **list** 或 **search** 请求，通过将 **PagedResultsRequest** 集设为 **abandonQuery**，值等同于在前一结果的 **PartialOutcomeQualifier** 中返回的 **queryReference** 值，DUA 可以在任何时候指明不需要更多的页了。将不请求或返回更多的页。它的实施依赖于各页何时被清除。

在做出 **queryReference** 或 **abandonQuery** 选择的情况下，新的请求和最初的信息在以下方面将是相同的：

— **ListArgument** 中 **SearchArgument** 或 **object** 中的 **baseObject** 将匹配于陈述和最初的请求；

— **pagedResults** 的 **queryReference** 子分量等同于在先前结果的 **PartialOutcomeQualifier** 中返回的 **queryReference** 值；

— **ServiceControls** 数据类型的选项分量将为陈述和最初的请求规定相同的选项；

— **operationProgress**（如果出现）对陈述和最初的请求都是一样的。

否则，将返回带问题 **invalidQueryReference** 的 **serviceError**。

注 3 — 如果在搜索请求之间 DIB 发生了变化，那么 DUA 可能无法见到这些变化的效果。它依赖于执行情况。

注 4 — 即使 DUA 开始一个新的列表或搜索操作，一个查询引用可以继续保持有效。一个 DUA 请求可以通过若干查询来请求分页结果，而后返回给一个早期的查询，并利用提供给它的查询引用请求下一页结果。DUA 能返回的“活动”查询引用数量是一个本地的 DSA 执行选项，是这些查询引用的生命周期。

注 5 — 支持 **abandonQuery** 选择仅适用于旧的第四版本系统。

注 6 — 当 DAP 关联终止时，对所有相关分页结果的访问丢失。分页结果只能在最初调用它们的 DAP 关联中进行访问。

7.10 安全性参数

SecurityParameters 用于管理与号码簿操作有关的各种安全特性的操作。

注 1 — 这些参数从发送方传送给接收方。操作变量中出现参数的请求方就是发送方，执行者为接收方。在结果中，角色相反。

```
SecurityParameters ::= SET {
    certification-path          [0]  CertificationPath          OPTIONAL,
    name                        [1]  DistinguishedName          OPTIONAL,
    time                        [2]  Time                       OPTIONAL,
    random                      [3]  BIT STRING                 OPTIONAL,
    target                      [4]  ProtectionRequest          OPTIONAL,
    response                    [5]  BIT STRING                 OPTIONAL,
    operationCode               [6]  Code                       OPTIONAL,
    attributeCertificationPath  [7]  AttributeCertificationPath OPTIONAL,
    errorProtection             [8]  ErrorProtectionRequest     OPTIONAL,
    errorCode                   [9]  Code                       OPTIONAL }

ProtectionRequest ::= INTEGER { none (0), signed (1) }
```

```
Time  ::=  CHOICE {
        utcTime                 UTCTime,
        generalizedTime         GeneralizedTime }

ErrorProtectionRequest  ::=  INTEGER { none (0), signed (1) }
```

CertificationPath 分量为一个包含签署方用户证书的序列，并且任选地，为一个或多个认证权威部门（CA）证书的序列。（见 ITU-T X.509 建议书 | ISO/IEC 9594-8 第 7 节。）用户证书用于绑定签署方的公共密钥和不同名称，并可以用来验证对请求变量、响应或错误的签名。如果签署了请求变量、响应或错误，那么该参数将出现，并包含签署方的用户证书。可以出现额外的证书，并可用来确定签署方的用户证书是否有效。如果接收方共用同一认证权威部门作为签署方，那么不需要额外的证书。如果接收方为确认需要一个认证途径，并且未出现一个可接受的参数，那么接收方是否拒绝签名或者尝试确定一个认证途径将是一个局部问题。

name 为变量或结果第一个计划中接收方的不同名称。例如，如果 DUA 产生一个经标记的变量，那么名称为操作提供给它的 DSA 的不同名称。

注 2 — 第一个计划的接收方有可选的、由内容区分的不同名称，**name** 可以是一个可选的名称。不过，如果不使用主要的不同名称，那么基于 **name** 值的认证和访问控制可能无法如期望的那样开展工作。

time 是计划中的终止时间，针对的是请求、响应或错误的有效性。它与随机数结合使用，使得能够检测重放攻击。

random 值应是一个不同于每个请求、响应或错误的数字。它与时间参数一起使用，以便能够检测重放攻击。如果要求序列完整性，那么随机变量可用于承载一个序列完整性数字，如下所述：

a） 与操作变量一起使用的随机值利用来自以下的预先商定序列（例如，前一个值加 1）获得：
   i） 对绑定中从系统发送的第一个操作，由远程对等系统在绑定操作变量/结果中传送的随机值；以及
   ii） 对后续操作，在相同方向的前一个操作中传送的随机值。

b） 与操作结果或错误一起使用的随机值利用来自请求中随机值的预先商定序列（例如，请求变量中的随机数加 1）获得：

target ProtectionRequest 只可出现于待完成的操作请求中，并指明请求方有关提供给结果的保护等级的优先权。提供了两个保护等级：**none**（没有请求保护，缺省值），以及 **signed**（请求号码簿对结果进行标记）。实际提供给结果的保护等级以结果形式指明，依据号码簿的限制，它可能等于或小于所请求的等级。

response 用于将任何信息传送回请求的发起者。

operationCode 用于将操作代码安全地绑定于请求变量、结果或错误。

attributeCertificationPath 用于为基于规则的访问控制传送一个安全许可证，或属性证书中的其他属性，确认属性证书所需的证书为可选。

errorProtection 只可出现于待完成的操作请求中，并指明请求方有关提供给任何错误的保护等级的优先权。提供了两个保护等级：**none**（没有请求保护，缺省值），以及 **signed**（请求号码簿对结果进行标记）。实际提供给错误的保护等级以错误形式指明，依据号码簿的限制，它可能等于或小于所请求的等级。

注 3 — DUA 可以要求任何安全标签正文都应利用正文选择返回一个属性值。

当响应一个操作返回一个错误时，**errorCode** 用于保护错误代码。

如果 **Time** 的语法选为 **UTCTime** 类型，那么 2 个数字表示的年字段的值将被解释为 4 个数字表示的年值，如下所示：

— 如果 2 个数字表示的值为 00- 49（包含），那么值将加上 2000。
— 如果 2 个数字表示的值为 90- 99（包含），那么值将加上 1900。

如果商定的版本为 **v2** 或更高版本，那么将使用 **GeneralizedTime**。当商定结果为 **v1** 时，**GeneralizedTime** 的使用可能有碍相互作用，实现方案无法感知是可能选择 **UTCTime** 还是选择 **GeneralizedTime**。这是那些用于规定域的版本的责任，在这些域中将使用号码簿规范，例如描述概貌组，什么时候可以使用 **GeneralizedTime**。**UTCTime** 将不用于描述任何超过 2049 年的日期。

7.11 访问控制程序的公共元素

当 **basic-access-control**、**rule-based-access-control** 或二者都起作用时，本小节用于定义所有抽象服务操作公用的程序元素。如果两种机制都起作用，那么其应用次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么来自其他机制的许可将不覆盖它。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

7.11.1 基本访问控制程序的公共元素

7.11.1.1 废弃别名

如果在定位目标对象条目（在抽象服务操作变量中确定）过程中要求废弃别名，那么为了废弃别名不需要特定的许可。不过，如果别名废弃将产生返回一个 **ContinuationReference**（即在 **Referral** 中），那么应用以下访问控制序列。如果 DSA 将请求链接至另一个 DSA，并从其处收回一个提名，那么若提名中的 **targetObject** 等同于链接请求中的 **targetObject**，则访问控制将适用于提名。也就是说，DSA 将对所有的提名进行监管，不论其是本地产生的还是远程产生的。

1） 别名条目需要取得读许可。如果未赋予许可，那么依据第 7.11.1 节中所述的程序，操作失败。

2） **aliasedEntryName** 属性及其所含的单个值需要取得读许可。如果未赋予许可，那么操作失败，并返回一个带问题 **aliasDereferencingProblem** 的 **nameError**。**matched** 元素将包含别名条目的名称。

注 — 除了上面所述的访问控制，安全策略还可以防止泄漏知识信息，否则它将作为 **Referral** 中的 **ContinuationReference** 予以传送。如果这样一个策略发挥作用，并且如果 DUA 通过规定 **chainingProhibited** 来约束服务，那么号码簿可能返回一个带问题 **chainingRequired** 的 **serviceError**。否则，将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**。

7.11.1.2 返回名称错误

如果在执行抽象服务操作时不能找到特定的目标对象（别名或条目）—例如，待读条目的名称或 **search** 请求中的 **baseObject**，那么将返回一个带问题 **noSuchObject** 的 **nameError**。**matched** 元素将包含下一个赋予了 *DiscloseOnError* 许可的上级条目的名称，或者 DIT 根的名称（即一个空的 **RDNSequence**）。

注 — 第二个选项可以通过 DSA 获得，它不访问所有的上级条目。

7.11.1.3 不透露条目是否存在

如果在 **rule-based-access-control** 下拒绝访问，那么 DiscloseOnError 许可不适用。

如果在执行抽象服务操作时特定的目标对象条目未赋予所需的条目级许可 — 例如，待读条目，那么操作失败，返回的错误为以下之一：如果目标条目赋予了 *DiscloseOnError* 许可，那么返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**；否则，返回一个带问题 **noSuchObject** 的 **nameError**。**matched** 元素将包含下一个赋予了 *DiscloseOnError* 许可的上级条目的名称，或者 DIT 根的名称（即一个空的 **RDNSequence**）。

注 — 第二个选项可以通过 DSA 获得，它不访问所有的上级条目。

另外，无论何时当号码簿检测到一个操作错误（包括提名），它将确保在返回该错误时不否认已命名目标条目及其任何上级的存在。例如，在返回一个带问题 **timeLimitExceeded** 的 **serviceError** 或带问题 **notAllowedOnNonLeaf** 的 **updateError** 之前，号码簿将确认 *discloseOnError* 许可赋予了目标条目。如果未赋予，那么接着执行上面段落中所述的程序。

7.11.1.4 返回不同的名称

在比较、列表或搜索操作中，如果是作为废弃一个别名的结果，那么 **object**（或 **baseObject**）条目需要取得 *ReturnDN* 许可，对象的不同名称将在操作结果的 **name** 参数中返回（见第 9.2.3 节）。如果未赋予该许可，那么号码簿将为条目返回一个别名，如第 7.7 节所述，或者将全部忽略 **name** 参数。

在读或搜索操作中，为了返回其在 **EntryInformation** 中的不同名称，条目需要取得 *ReturnDN* 许可。如果未赋予该许可，那么号码簿将返回一个别名，如第 7.7 节所述，或者如果没有任何别名可用，那么操作将失败，产生一个 **nameError** 错误（在读操作情况下），或者从结果中忽略该条目（在搜索操作情况下）。

如果在结果中返回用户提供的别名，那么不得将 **CommonResults** 的 **aliasDeferenced** 标志设为 **TRUE**。

7.11.2 基于规则的访问控制程序的公共元素

7.11.2.1 访问条目（条目级许可）

为了访问一个条目，需要得到至少访问条目中一个属性值的许可。如果未赋予条目级的许可，那么将返回带问题 **noSuchObject** 的 **nameError**。

7.11.2.2 返回条目名称

为了返回一个条目的 DN，需要允许访问至少一个条目 RDN 正文变量的所有属性值（术语定义为 *RDN* 许可）。对条目的任何上级没有任何许可要求。如果未赋予 RDN 许可，那么 DSA 可以选择为已赋予 RDN 许可的属性值返回一个条目有效别名的 DN，或者从操作结果中删去名称分量。

注 — 有关适当别名的选择将在第 7.7 节的注释中做进一步描述。

7.11.2.3 废弃别名

为了废弃别名，需要得到访问 **aliasedEntryName** 属性值的许可。

7.11.2.4 返回名称错误（noSuchObject）

带问题 **noSuchObject** 的 **nameError** 的 **matched** 分量将设为下一个上级条目的名称，其请求方拥有 RDN 许可。如果这样一个条目不适用于产生错误的 DSA，那么将返回 DIT 根的名称。

7.11.2.5 访问属性

为了访问一个属性，需要得到至少访问属性中一个值的许可。

7.11.2.6 删除信息

为了删除一个属性值，需要取得访问该值的许可。当删除一个条目或一个属性时，如果至少删除了一个属性值，那么操作将返回一个成功响应，而不管请求删除多少值。

7.11.2.7 调用搜索规则

为了依据搜索操作的变量对搜索规则进行评估，发起搜索操作的请求方需要调用搜索规则许可。为了访问搜索规则属性或包含它的分条目，用户不需要任何其他许可。

7.11.3 族信息

族信息当作为任何其他信息，除了 ACI，其 **ProtectedItem** 标记为 **includeFamily**；如果 ACI 适用于祖先或族成员，那么这将引起下属族成员受制于同一 ACI。只有当应用于 **entry** 保护项时，**IncludeFamily** 才有意义。

7.12 管理DSG信息树

由 DSA 持有的 DSA 信息树可以利用号码簿抽象服务进行管理。当管理 DSA 信息树时：

— DSA 中的所有 DSE 通过 DAP 都是可见的，包括根 DSE；

— 属性定义为任何用户修改都不可以被修改（虽然如果它不支持请求的修改，DSA 可以利用带问题 **unwillingToPerform** 的 **serviceError** 进行回复）；

— 知识只是另一个可以读和修改的属性；以及

— DSA 从不链接请求或返回提名或连续引用。

DSE 的可见性和操作属性的检索或修改可以通过正常方式的访问控制进行控制。

DSA 信息树的管理通过使用以下程序的 DUA 实现：

1） DUA 直接绑定于持有 DSA 信息树的 DSA 上，将要对之进行管理；

2） 对每个用于管理 DSA 信息树的操作：

— 将设置 **manageDSAIT** 扩展位；

— 将设置 **manageDSAIT** 选项；

— 如果需要管理特定的复制平面，那么将包括 **manageDSAITPlaneRef** 选项；

号码簿忽略以下分量：

— **CommonArgument** 中的 **operationProgress**；

— **CommonArgument** 中的 **referenceType**；

— **CommonArgument** 中的 **entryOnly**；

— **CommonArgument** 中的 **nameResolveOnMaster**；以及

— **ServiceControls** 中的 **chainingProhibited**。

7.13 条目族程序

如第 7.3.2 节所规定，出于操作评估目的，可以将复合条目内的族成员组合在一起。这种组合只与比较、搜索和移去条目操作有关。如果族组合是为任何其他操作规定的，那么忽略之。

为了依据 **entryInformationSelection** 的 **familyReturn** 分量确定将返回哪些族成员，引入了*起作用的成员*和*参与成员*两个概念。这些概念只与将返回条目信息的操作相关，即读、搜索和修改条目操作。

如果族成员对操作评估起积极的作用，那么它标记为起作用的成员。如果它是匹配过滤器的族组的一部分，并且如果它持有一个或多个匹配于非求反过滤器项的属性，那么族成员对匹配起作用。如果它持有某个给定类型的属性，并且如果相同类型的求反过滤器项不匹配，那么它也起作用。在读或修改条目操作中，只有操作选择的族成员（如操作的 **object** 分量所规定）才能被标记为起作用的成员和参与成员。在搜索操作中，族组合针对的是过滤器匹配。如果族组匹配过滤器（见第 7.8.3 节），那么所有对匹配起积极作用的成员都将被标记为起作用的成员，而组的所有条目都被标记为参与成员。如果所用的过滤器是默认的过滤器（**and : { }**），那么族组的所有成员都将被标记为参与成员，但不被标记为起作用的成员。

当复合条目的族组匹配过滤器并且 **SearchArgument** 规定了层次型选择（除了 **self**）时，合适的话，对所选的条目也做标记。如果复合条目的祖先标记为参与成员（也有可能标记为起作用的成员），那么将选择不是复合条目的、层次型组的所有引用条目，否则将之排除在外。如果引用的条目是一个复合条目，那么按以下所述对其成员进行标记。以相同的方式对与匹配复合条目成员拥有相同局部成员名称的引用复合条目的每个成员进行标记。对引用复合条目的所有其他成员都不做标记。

由于一个搜索过滤器可能匹配若干个复合条目，因此最终的选择和标记将是单个匹配复合条目选择和标记的联合。

如果一个不是复合条目的匹配条目在其层次型选择中引用了一个复合条目，那么该复合条目的所有成员都被标记为参与成员。

有关该条目标记如何影响条目信息的返回在第 7.6.4 节中详述。

族成员可以包装进一个 **family-information** 派生的属性。如果在结果中只返回了一个单个复合条目成员，那么将不执行包装。不过，如果从读或修改条目操作返回了若干成员，那么将对这些成员进行包装。当一个搜索操作返回若干复合属性成员时，将对它们进行包装，除非设置了 **separateFamilyMembers** 搜索控制选项，在这种情况下，成员将作为单独条目返回。

当执行涉及复合条目的搜索操作时，对搜索操作有四个相关的阶段：

a） 每个感兴趣条目中的族成员组，如 **familyGrouping** 所定义，在逻辑上在每个候选条目中考虑（即通过子集选择）。通过将所有的组属性汇集在一起，认为给定属性类型的所有属性值属于这个单个属性类型，即使它们来自不同的族成员。

b） 过滤器适用于每个族组；如果过滤器满足组要求，那么复合条目满足过滤器要求，并考虑通过过滤器进行选择。对族成员进行标记，如上所述。

c） 增加标记条目，通过 **EntryInformationSelection** 中的 **familyReturn** 来规定，以标记将返回的所有条目。

d） 如果在管理搜索规则中出现 **additionalControl** 分量（见 ITU-T X.501 建议书| ISO/IEC 9594-2 第 16.10.8 节），那么改变标记以及因此而返回的内容，作为处理所引用控制属性的结果。

8 绑定和解开操作

号码簿绑定和号码簿解开操作分别在第 8.1 节和第 8.2 节中定义，由 DUA 在访问号码簿的某个特定周期的开始和结束之时使用。

8.1 号码簿绑定

8.1.1 号码簿绑定语法

号码簿绑定操作在访问号码簿的周期的开始使用。操作参数可以由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书| ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

```
directoryBind  OPERATION  ::= {
        ARGUMENT          DirectoryBindArgument
        RESULT            DirectoryBindResult
        ERRORS            { directoryBindError } }

DirectoryBindArgument  ::=  SET  {
        Credentials   [0]     Credentials OPTIONAL,
        Versions      [1]     Versions DEFAULT {v1} }

Credentials  ::=  CHOICE  {
        simple                [0]       SimpleCredentials,
        strong                [1]       StrongCredentials,
        externalProcedure     [2]       EXTERNAL,
        spkm                  [3]       SpkmCredentials,
        sasl                  [4]       SaslCredentials }

SimpleCredentials  ::=  SEQUENCE  {
        Name          [0]     DistinguishedName,
        Validity      [1]     SET {
                time1         [0]       CHOICE {
                        utc               UTCTime,
                        gt                GeneralizedTime } OPTIONAL,
                time2         [1]       CHOICE {
                        utc               UTCTime,
                        gt                GeneralizedTime } OPTIONAL,
                random1       [2]       BIT STRING  OPTIONAL,
                random2       [3]       BIT STRING  OPTIONAL } OPTIONAL,
        password    [2]       CHOICE {
                unprotected               OCTET STRING,
                protected                 SIGNATURE {OCTET STRING} } OPTIONAL}

StrongCredentials  ::=  SET  {
        certification-path              [0]     CertificationPath OPTIONAL,
        bind-token                      [1]     Token,
        name                            [2]     DistinguishedName OPTIONAL,
        attributeCertificationPath      [3]     AttributeCertificationPath OPTIONAL }

SpkmCredentials  ::=  CHOICE  {
        req           [0]     SPKM-REQ,
        rep           [1]     SPKM-REP-TI }
```

```
SaslCredentials  ::=  SEQUENCE {
        Mechanism           [0]    DirectoryString { ub-saslMechanism },
        credentials         [1]    OCTET STRING OPTIONAL,
        saslAbort           [2]    BOOLEAN DEFAULT FALSE }

Token  ::=  SIGNED  { SEQUENCE {
        algorithm                   [0]     AlgorithmIdentifier,
        name                        [1]     DistinguishedName,
        time                        [2]     Time,
        random                      [3]     BIT STRING,
        response                    [4]     BIT STRING OPTIONAL,
        bindIntAlgorithm            [5]     SEQUENCE SIZE (1..MAX) OF AlgorithmIdentifier OPTIONAL,
        bindIntKeyInfo              [6]     BindKeyInfo OPTIONAL,
        bindConfAlgorithm           [7]     SEQUENCE SIZE (1..MAX) OF AlgorithmIdentifier OPTIONAL,
        bindConfKeyInfo             [8]     BindKeyInfo OPTIONAL } }

Versions  ::=  BIT STRING {v1(0), v2(1) }

DirectoryBindResult  ::=  DirectoryBindArgument

directoryBindError  ERROR  ::=  {
        PARAMETER                OPTIONALLY-PROTECTED {
                SET {
                        versions                [0]     Versions DEFAULT {v1},
                        error                           CHOICE {
                                serviceError    [1]     ServiceProblem,
                                securityError   [2]     SecurityProblem } } } }

BindKeyInfo  ::=  ENCRYPTED  { BIT STRING }
```

8.1.2 号码簿绑定变量

DirectoryBindArgument 的 **credentials** 变量允许号码簿建立用户的身份。证书可以是 **simple** 或 **strong** 或外部定义的（**externalProcedure**）（如 ITU-T X.509 建议书| ISO/IEC 9594-8 中所述）。

如果使用 **simple**，那么它包括一个 **name**（总为对象的不同名称）、一个可选的 **validity** 以及一个可选的 **password**。这提供了有限程度的安全。**password** 可以是 **unprotected** ，或者可以是 **protected**（保护 1 或保护 2），如 ITU-T X.509 建议书| ISO/IEC 9594-8 中所述。**validity** 提供了 **time1**、**time2**、**random1** 和 **random2** 变量，其含义来自双边协议，可用于检测重放。在某些情况下，受保护的口令可以通过一个对象来检查，该对象只有在局部重建对其自身口令拷贝的保护并对结果与绑定变量（**password**）中的值进行比较后，才能知晓口令。在其他情况下，可能直接进行比较。

如果商定的版本为 **v2** 或更高版本，那么将对 **time1** 和 **time2** 使用 **GeneralizedTime**。当商定结果为 **v1** 时，**GeneralizedTime** 的使用可能有碍相互作用，实现方案无法感知是可能选择 **UTCTime** 还是可能选择 **GeneralizedTime**。这是那些用于规定域的版本的责任，在这些域中将使用号码簿规范，例如描述概貌组，什么时候可以使用 **GeneralizedTime**。**UTCTime** 将不用于描述超过 2049 年的日期。

如果使用 **strong**，那么它包括一个 **bind-token**、一个可选的 **certification-path**（认证和认证权威部门交叉认证序列，如 ITU-T X.509 建议书| ISO/IEC 9594-8 中所述）以及请求方的 **name**。这使得号码簿能够对建立关联的请求方身份进行验证，反之亦然。如果在绑定操作中使用 **StrongCredentials** 或 **SpkmCredentials**，那么传送有关身份和验证的信息。这使得能够对任何一个实体的身份进行验证，还使得能够使用已经建立的密码以及完整性密码密钥资料。

BindIntAlgorithm 和 **bindConfAlgorithm** 分量用于商定密码算法，以便用于保护绑定中的后续操作。请求方包括一个按优先次序排列的支持算法清单。号码簿从清单中选择一个算法，它符合其自身的安全策略要求，并在响应中指明这一点。

完整性和机密性算法使用的会话密钥通过使用 **bindIntKeyInfo** 和 **bindConfKeyInfo** 字段来建立。通过产生一个适当长度的会话密钥，并用其他公共密钥进行加密，请求方和号码簿对会话密钥的选择都有影响。会话密钥是这两个分量的异或。注意，请求方可以将会话密钥的生成交给号码簿，在这种情况下，将从绑定变量中删去上述各字段。

注 1 — 认证证书可能通过安全交换服务元素进行传送（见 ITU-T X.519 建议书| ISO/IEC 9594-5），在这种情况下，它们将不出现在绑定变量或结果中。

如果对操作进行标记和加密，那么包含属性的属性证书（见 ITU-T X.509 建议书 | ISO/IEC 9594-8 第 12 节）可以用于传送属性访问所需的许可证。**attributeCertificationPath** 用于传送基于规则的访问控制的安全许可证，或者在属性证书中传送的其他属性，可选地，还有验证属性证书所需的证书。

绑定令牌的变量如下使用。**algorithm** 是用于标记该信息的算法标识符。**name** 是计划中接收方的名称。**time** 参数包含令牌的终止时间。**random** 数是一个应不同于各个未终止令牌的数，接收方可用之来检测重放攻击。

注 2 — 当名称用在简单或增强证书中时，如果存在，可能使用可选的不同名称。不过，如果不使用主要的不同名称，那么基于名称的认证和访问控制可能无法如期望的那样开展工作。在成功处理经认证的 BIND 操作后，在 BIND 变量中无论使用什么名称，各绑定实体相互间都将知道其主要的不同名称，以便在 BIND 起作用的情况下推动访问控制操作。

如果使用 **externalProcedure**，那么正在使用的认证方案的语义将在号码簿规范范围之外。

当使用 RFC 2222 中规定的简单认证和安全层（SASL）时，使用 **sasl**。如果通过值设为空字符串的 **SaslCredentials** 机制来调用 **directoryBind** 操作，那么将返回一个 **inappropriateAuthentication** 的 **SecurityError**。

DirectoryBindArgument 的 **versions** 变量用于确定 DUA 准备参与的服务的版本。值 **v1** 表示协议版本 1，值 **v2** 表示协议版本 2。如果在后续 **ModifyEntry** 操作中将传送 **alterValues** 或 **resetValue** 修改类型，或者需要一个非 **NULL** 的结果（见第 11.3 节），那么将使用值 **v2**。如果对增加条目、移去条目、修改条目、修改 DN 使用错误或结果标记，那么值将设为 **v2**。

通过以下措施来推动向号码簿未来版本的迁移：

a） 将接受和忽略 **DirectoryBindArgument** 的任何元素，而非本号码簿规范中定义的那些元素；

b） 将接受和忽略有关未定义的 **DirectoryBindArgument** 命名位（如版本）的各额外选项。

如果要求抢占响应认证，那么 **response** 分量用于承载一个随机数。

BindIntAlgorithm、**bindKeyInfo**、**bindConfAlgorithm** 和 **bindConfKey** 分量用于承载保护绑定中后续操作所需的信息。

8.1.3 号码簿绑定结果

如果绑定请求成功，那么将返回一个结果。

DirectoryBindResult 的 **credentials** 变量允许用户建立号码簿的身份。它允许将用于确定 DSA（直接提供号码簿服务）的信息传送给 DUA。其形式（即 **CHOICE**）将同于用户提供的形式。

DirectoryBindResult 的 **versions** 参数用于指明 DSA 将实际提供哪个版本的服务（DUA 请求的）。

8.1.4 号码簿绑定错误

如果绑定请求失败，那么将返回一个绑定错误。

directoryBindError 的 **versions** 参数指明 DSA 支持哪个版本。

securityError 或 **serviceError** 将按如下方式提供：

— **securityError** **inappropriateAuthentication**
invalidCredentials
blockedCredentials

— **serviceError** **unavailable**
saslBindInProgress

8.2 号码簿解开

访问号码簿周期结束之时的解开针对的是 ITU-T X.519 建议书| ISO/IEC 9594-5 第 7.6.4 节中所规定的 OSI 环境以及 ITU-T X.519 建议书| ISO/IEC 9594-5 第 9.3.2 节中所规定的 TCP/IP 环境。

注 — 在解开时，尚未访问的所有分页结果将变得不可访问，应去除。

9 号码簿读操作

有两个“类似读”操作：**read** 和 **compare**，分别在第 9.1 节和第 9.2 节中定义。为方便起见，在第 9.3 节中定义的 **abandon** 操作将与这些操作结合在一起。

9.1 读

9.1.1 读语法

读操作用于从一个明确确定的条目中提取信息。它还可以验证不同的名称。操作变量可以由请求方进行标记（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记。

```
read  OPERATION  ::= {
        ARGUMENT          ReadArgument
        RESULT            ReadResult
        ERRORS            { attributeError | nameError | serviceError | referral | abandoned |
                          securityError }
        CODE              id-opcode-read }

ReadArgument  ::=  OPTIONALLY-PROTECTED {
        SET {
              object                    [0]   Name,
              selection                 [1]   EntryInformationSelection  DEFAULT { },
              modifyRightsRequest       [2]   BOOLEAN DEFAULT FALSE,
              COMPONENTS OF                   CommonArguments } }

ReadResult  ::=  OPTIONALLY-PROTECTED {
        SET {
              entry               [0]   EntryInformation,
              modifyRights        [1]   ModifyRights OPTIONAL,
              COMPONENTS OF             CommonResults } }

ModifyRights  ::=  SET OF SEQUENCE {
        item                        CHOICE {
                    entry                 [0]   NULL,
                    attribute             [1]   AttributeType,
                    value                 [2]   AttributeValueAssertion },
        permission                        [3]   BIT STRING { add (0), remove (1), rename (2), move (3) } }
```

9.1.2 读变量

object 变量用于确定自其请求信息的对象条目。如果 **Name** 涉及一个或多个别名，那么废弃它们（除非相关的服务控制禁止之）。**Name** 可以是一个可选的名称，并可包括正文信息，如 ITU-T X.501 建议书| ISO/IEC 9594-2 第 9.3 节中所述。

selection 变量指明自条目请求什么信息（见第 7.6 节）。不过，不应假设返回的属性等同于请求的那些属性或限于请求的那些属性。

CommonArguments（见第 7.3 节）包括有关服务控制和适用于请求的安全参数的规定。出于本操作的目的，**sizeLimit** 分量不相关，如果提供，将被忽略。如果请求方对该操作的变量进行标记、加密或者标记和加密，那么 **SecurityParameters**（见第 7.10 节）分量将包括在变量中。

modifyRightsRequest 变量用于请求将请求方的修改权限返回给条目及其属性。

9.1.3 读结果

如果请求成功，那么将返回结果。

条目结果参数持有请求的信息（见第 7.7 节）。如果因 **EntryInformationSelection** 中 **familyReturn** 元素出现而提出要求，那么这可以包括族信息。

如果通过 **modifyRightsRequest** 变量提出请求，那么 **modifyRights** 参数出现，用户对某些或所有的请求条目信息拥有修改特权，局部安全策略允许返回该信息。如果返回，那么为条目和 **selection** 变量中规定的属性返回请求方的修改权限。参数包含以下内容：

— 为 **entry**、对每个请求的用户 **attribute**（用户拥有增加或移去权限）、为每个返回的属性 **value**（用户增加或移去它的权限不同于对应属性的那些权限）返回一个 **SET** 的元素。

— 返回的 **permission** 指明用户在条目上实施的哪些操作或行为将取得成功。在一个条目的情况下，**remove** 指明 **RemoveEntry** 操作将取得成功；**rename** 指明，如果 **newSuperior** 参数不存在，那么 **ModifyDN** 将取得成功；**move** 指明，如果 **newSuperior** 参数出现，那么 **ModifyDN** 和未改变的 RDN 将取得成功。

在属性和值的情况下，**add** 指明增加属性或值的 **ModifyEntry** 将取得成功；**remove** 指明，移去属性或值的 **ModifyEntry** 将取得成功。

注 — 将条目移至一个新上级的操作还可能依赖于与新上级相关的许可（例如，通过 **basic-access-control**）。当确定 **permission** 时，这些将被忽略。

CommonResults（见第 7.4 节）包括适用于响应的安全参数。如果号码簿对结果进行标记、加密或者标记和加密，那么 **SecurityParameters** 分量（见第 7.10 节）将包括在结果中。

9.1.4 读错误

如果请求失败，那么将报告其中的一个列出错误。如果无法返回任何一个明确列于 **selection** 中的属性，那么将报告一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。将报告其他错误的情况在第 12 节中定义。

9.1.5 基本访问控制的读操作决策点

如果 **rule-based-access-control** 也应用，那么有关 **basic-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **basic-access-control** 对正在读的条目起作用，那么应用以下访问控制序列：

1） 对正在读的条目需要读许可。如果未赋予许可，那么依据第 7.11.1.3 节，操作失败。

2） 如果 **selection** 的 **infoTypes** 元素规定只能返回属性类型，那么对每个将要返回的属性类型，需要读许可。如果未赋予许可，那么从 **ReadResult** 中删去属性类型。如果作为应用这些控制的结果，没有返回任何属性信息，那么依据第 9.1.5.1 节，整个操作失败。

3） 如果 **selection** 的 **infoTypes** 元素规定返回属性类型和值，那么对每个将要返回的属性类型和值，需要读许可。如果对属性类型未赋予许可，那么从 **ReadResult** 中删去属性。如果对属性值未赋予许可，那么从其对应的属性中删去值。在未将许可赋予属性内任何值的情况下，返回一个包含空 **SET OF AttributeValue** 的 **Attribute** 元素。如果作为应用这些控制的结果，没有返回任何属性信息，那么依据第 9.1.5.1 节，整个操作失败。

注 — 允许 DAP 读操作的特权在 LDAP 环境中可能不起作用，当中需要得到浏览许可，以便支持相当的读服务。

9.1.5.1 错误返回

如果操作失败，如第 9.1.5 节 2）和 3）定义，那么有效的错误返回为以下之一：

a） 如果规定了一个无限制的选项（即 **allUserAttributes** 或 **allOperationalAttributes**），那么将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**。

b） 否则，如果规定了一个 **select** 选项（在 **attributes** 中与/或在 **extraAttributes** 中），那么如果为任何选定的属性赋予了 *DiscloseOnError* 许可，那么将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**。否则，将返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

9.1.5.2 不透露不完整的结果

如果在 **EntryInformation** 中返回一个不完整的结果，即因适用的访问控制而删去了某些属性或属性值，那么 **incompleteEntry** 元素将被设为 **TRUE**，条件是 *DiscloseOnError* 许可赋给至少一个限制在结果外的属性类型，或者赋给至少一个限制在结果外的属性值（对该属性类型赋予了读许可）。

9.1.6 基于规则的访问控制的读操作决策点

如果 **basic-access-control** 也应用，那么有关 **rule-based-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **rule-based-access-control**、**rule-and-basic-access-control** 或 **rule-and-simple-access-control** 对正在读的条目起作用，那么应用以下访问控制：

1） 如果在 **rule-based-access-control** 下拒绝条目级访问，那么依据第 7.11.2.4 节，操作失败，返回一个带问题 **noSuchObject** 的 **nameError**。

2） 如果在 **basic-access-control** 方案下不允许访问条目，如第 9.1.5 节 1）所述，那么依据第 7.11.1.3 节，操作失败。

3） 如果 **selection** 的 **infoTypes** 元素规定只返回属性类型，那么如果在 **rule-based-access-control** 下，未准予访问该类型的所有属性值，那么从 **ReadResult** 中删去属性类型。如果作为应用这些控制的结果，未返回任何属性信息，那么依据第 9.1.5.1 节 b），整个操作失败，返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

4） 如果 **selection** 的 **infoTypes** 元素规定只返回属性类型，那么应用 **basic-access-control**，如第 9.1.5 节 2）所述。

5） 在 **rule-based-access-control** 下，如果 **selection** 的 **infoTypes** 元素规定返回属性类型和值，那么对每个将要返回的属性值，将准予访问。如果未准予访问某个属性值，那么从其对应的属性中删去该属性值。在未准予访问某个属性中任何属性值的情况下，从 **ReadResult** 中删去整个属性。如果作为应用这些控制的结果，未返回任何属性信息，那么整个操作失败，返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

6） 应用 **basic-access-control**，如第 9.1.5 节 3）所述。

7） 按第 7.11.2.2 节定义确定在操作结果中返回的名称。

9.2 比较

9.2.1 比较语法

比较操作用于一个值（作为请求变量提供）与某个特定对象条目中某个特定属性类型值的比较。操作变量可以由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

除了 **multiStrand**，可以使用任何 **familyGrouping** 值，所有成组族成员的属性都将在与假设的属性值命题的比较中使用。如果 **familyGrouping** 规定了 **multiStrand**，那么采用 **compoundEntry**。

```
compare OPERATION ::= {
        ARGUMENT        CompareArgument
        RESULT          CompareResult
        ERRORS          { attributeError | nameError | serviceError | referral | abandoned |
                        securityError }
        CODE            id-opcode-compare }
```

```
CompareArgument ::= OPTIONALLY-PROTECTED {
        SET {
              object                    [0]     Name,
              purported                 [1]     AttributeValueAssertion,
              COMPONENTS OF                     CommonArguments } }

CompareResult ::= OPTIONALLY-PROTECTED {
        SET {
              name                                Name  OPTIONAL,
              matched                             [0]       BOOLEAN,
              fromEntry                           [1]       BOOLEAN DEFAULT TRUE,
              matchedSubtype                      [2]       AttributeType OPTIONAL,
              COMPONENTS OF                       CommonResults } }
```

9.2.2 比较变量

object 变量为所考虑的特殊对象条目的名称。如果 **Name** 涉及一个或多个别名，那么废弃它们（除非相关的服务控制禁止之）。**Name** 可以是一个可选的名称，并可包括正文信息，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 9.3 节中所述。

purported 变量用于确定将要与条目中属性类型和值进行比较的属性类型和值。如果条目持有假设的属性类型或其子类型之一，或者存在一个为假设的属性类型或其子类型之一的共同条目属性（见第 7.6 节），并且如果存在一个匹配假设值的属性值（利用属性的 **equality** 匹配规则），那么比较结果为 TRUE。

注 — **compare** 请求无法满足变量中所规定的属性类型的友好属性类型的要求。

如果属性值命题中包括正文命题，那么将只对那些满足所有给定正文命题要求的值尝试匹配，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 8.9.2 节所述。如果在属性值命题中未包括任何正文命题，那么将应用缺省正文命题，如 ITU-T X.501 建议书| ISO/IEC 9594-2 第 8.9.2.2 节所述。

CommonArguments（见第 7.3 节）包括有关服务控制和适用于请求的安全参数的规定。出于该操作的目的，**sizeLimit** 分量不相关，如果提供，将被忽略。如果请求方对该操作的变量进行标记、加密或者标记和加密，那么 **SecurityParameters**（见第 7.10 节）分量将包括在变量中。

9.2.3 比较结果

如果请求成功（即真实地完成了比较），那么将返回结果。

name 为条目的不同名称或条目的别名，如第 7.7 节所述。只有当别名废弃时、当 RDN 已解析为主要 RDN 时，或者当内容选择已应用时、当将要返回的名称不同于操作变量中所提供的 **Object** 名称时，它才出现。

matched 结果参数持有比较结果。如果对值进行了比较并匹配，那么参数取 **TRUE** 值，如果不是这样，那么取 **FALSE** 值。

如果 **fromEntry** 为 **TRUE**，那么信息与条目进行比较；如果为 **FALSE**，那么信息与拷贝进行比较。

只有当匹配结果为 TRUE 并且因假设属性的子类型匹配而使匹配取得成功时，**matchedSubtype** 参数才出现。如果多个这样的子类型可用，那么返回层次中最高的那个。

CommonResults（见第 7.4 节）包括适用于响应的安全参数。如果号码簿对结果进行标记、加密或者标记和加密，那么 **SecurityParameters** 分量（见第 7.10 节）将包括在结果中。

9.2.4 比较错误

如果请求失败，那么将报告其中一个列出的错误。对将报告特殊错误的情况在第 12 节中进行定义。

9.2.5 基本访问控制的比较操作决策点

如果 **rule-based-access-control** 也应用，那么有关 **basic-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **basic-access-control** 对正在比较的条目起作用，那么应用以下访问控制序列：

1） 对将要比较的条目需要读许可。如果不赋予该许可，那么依据第 7.11.1.3 节，该操作失败。

2） 对正在比较的属性需要比较许可。如果不赋予该许可，那么依据第 9.2.5.1 节，该操作失败。

3） 如果在正在比较的属性中存在一个匹配 **purported** 变量的值，并且赋予了比较许可，那么操作在 **CompareResult** 的 **matched** 结果参数中将返回值 **TURE**。否则，操作将返回值 **FALSE**。

9.2.5.1 错误返回

如果操作失败，如第 9.2.5 节 2）定义，那么有效的错误返回是如下之一：如果将 *DiscloseOnError* 许可赋予了正在比较的属性，那么将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**；否则，将返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

9.2.6 基于规则的访问控制的比较操作决策点

如果 **basic-access-control** 也应用，那么有关 **rule-based-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **rule-based-access-control**、**rule-and-basic-access-control** 或 **rule-and-simple-access-control** 对正在比较的条目起作用，那么应用以下访问控制：

1） 如果在 **rule-based-access-control** 下拒绝条目级访问，那么依据第 7.11.2.4 节，操作失败，返回一个带问题 **noSuchObject** 的 **nameError**；

2） 如果在 **based-access-control** 方案下不允许访问条目，如第 9.2.5 节 1）所述，那么依据第 7.11.1.3 节，操作失败；

3） 如果访问未赋予正在比较的属性值，那么号码簿将按以下方式开展工作，即仿佛属性值没有出现；

4） 应用 **basic-access-control**，如第 9.2.5 节 2）和 3）所述；

5） 按第 7.11.2.2 节定义确定在操作结果中返回的名称。

9.3 放弃

如果用户对结果不再感兴趣，那么可以利用 **abandon** 操作放弃查询号码簿的操作。操作变量可以由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书| ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

```
abandon  OPERATION  ::= {
        ARGUMENT          AbandonArgument
        RESULT            AbandonResult
        ERRORS            { abandonFailed }
        CODE              id-opcode-abandon }

AbandonArgument  ::=  OPTIONALLY-PROTECTED-SEQ {
        SEQUENCE  {
                invokeID              [0]    InvokeId } }

AbandonResult  ::=  CHOICE  {
        null                  NULL,
        information           OPTIONALLY-PROTECTED-SEQ {
                                      SEQUENCE {
                                            invokeID                InvokeId,
                                            COMPONENTS OF           CommonResultsSeq } } }
```

有一个单个变量 **invokeID**，用于确定将要放弃的操作。**InvokeID** 的值等于用于调用将要放弃的操作的 **invokeID**。

如果请求成功，那么将返回一个结果。如果号码簿对该结果进行标记、加密或者标记和加密，那么 **CommonResultsSeq**（见第 7.4 节）的 **SecurityParameters** 分量（见第 7.10 节）将包括在结果中。如果号码簿不对该操作的结果进行标记，那么不随结果传送任何信息。最初的操作将因 **abandoned** 错误而失败。

如果请求失败，那么将报告 **abandonFailed** 错误。作为一个局部问题，DSA 可以选择不放弃操作，而后返回 **abandonFailed** 错误。该错误在第 12.3 节中进行描述。

放弃仅适用于查询操作，即读、比较、列表和搜索操作。

DSA 可以在本地放弃放弃一个操作。如果 DSA 已将操作链接或多点传送至其他 DSA，那么它可以依次对它们进行查询，以便放弃操作。

10 号码簿搜索操作

有两个“类搜索”操作：列表和搜索，分别在第 10.1 节和第 10.2 节中定义。

10.1 列表

10.1.1 列表语法

列表操作用于获取一个明确确定的条目的直接下属。在某些情况下，返回的列表是不完整的。操作变量可以由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

```
list    OPERATION ::= {
        ARGUMENT        ListArgument
        RESULT          ListResult
        ERRORS          { nameError | serviceError | referral | abandoned | securityError }
        CODE            id-opcode-list }

ListArgument ::= OPTIONALLY-PROTECTED {
        SET {
                object                  [0]     Name,
                pagedResults            [1]     PagedResultsRequest OPTIONAL,
                listFamily              [2]     BOOLEAN DEFAULT FALSE,
                COMPONENTS OF                   CommonArguments } }

ListResult ::= OPTIONALLY-PROTECTED {
        CHOICE {
                listInfo                        SET {
                        name                                    Name  OPTIONAL,
                        subordinates            [1]             SET OF SEQUENCE {
                                rdn                                     RelativeDistinguishedName,
                                aliasEntry                      [0]     BOOLEAN DEFAULT  FALSE,
                                fromEntry                       [1]     BOOLEAN DEFAULT  TRUE },
                        partialOutcomeQualifier [2]             PartialOutcomeQualifier OPTIONAL,
                        COMPONENTS OF                           CommonResults },
                uncorrelatedListInfo    [0]     SET OF ListResult } }

PartialOutcomeQualifier ::= SET {
        limitProblem                    [0]     LimitProblem  OPTIONAL,
        unexplored                      [1]     SET SIZE (1..MAX) OF ContinuationReference  OPTIONAL,
        unavailableCriticalExtensions   [2]     BOOLEAN DEFAULT  FALSE,
        unknownErrors                   [3]     SET SIZE (1..MAX) OF ABSTRACT-SYNTAX.&Type  OPTIONAL,
        queryReference                  [4]     OCTET STRING  OPTIONAL,
        overspecFilter                  [5]     Filter OPTIONAL,
        notification                    [6]     SEQUENCE SIZE (1..MAX) OF Attribute  OPTIONAL,
        entryCount                              CHOICE {
                bestEstimate            [7]     INTEGER,
                lowEstimate             [8]     INTEGER,
                exact                   [9]     INTEGER } OPTIONAL,
        streamedResult                  [10]    BOOLEAN DEFAULT  FALSE }

LimitProblem ::= INTEGER {
        timeLimitExceeded (0), sizeLimitExceeded (1), administrativeLimitExceeded (2) }
```

10.1.2 列表变量

object 变量用于确定对象条目（或可能是根），将列出其直接下属。如果 **Name** 涉及一个或多个别名，那么废弃它们（除非相关的服务控制禁止这么做）。**Name** 可以是一个可选的名称，并可以包括正文信息，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 9.3 节所述。

pagedResults 变量用于请求逐页返回的操作结果，如第 7.9 节所述。

如果 **listFamily** 为 **TRUE**，并且 **object** 为祖先，那么列出的下属取自直接的下属族成员；不包括任何其他下属。否则，列出的下属只能取自不是族成员的直接下属。

CommonArguments（见第 7.3 节）包括适用于请求的服务控制规定。如果请求方将对该操作的变量进行标记、加密或者标记和加密，那么 **SecurityParameters** 分量（见第 7.10 节）将包括在变量中。

10.1.3 列表结果

依据访问控制，如果找到了 **object**，而不管是否返回下属信息，那么请求成功。

name 为条目的不同名称或条目的别名，如第 7.7 节所述。只有当别名废弃时、当 RDN 已解析为主要 RDN 时，或者当内容选择已应用时、当将要返回的名称不同于操作变量中所提供的 **object** 名称时，它才出现。

如果有的话，**subordinates** 参数用于传送命名条目直接下属上的信息。如果任何下属条目为别名，那么它们将不被废弃。

rdn 参数为下属的相对不同名称。这可能受第 7.7 节中为 **Name** 所述的正文影响。

fromEntry 参数用于指明信息是取自条目（**TRUE**）还是条目的一个拷贝（**FALSE**）。

aliasEntry 参数用于指明下属条目是一个别名条目（**TRUE**）还是不是一个别名条目（**FALSE**）。

partialOutcomeQualifier 由九个如下所述的子分量组成。无论何时，当由于时间限制、大小限制或管理限制等问题、由于未开发 DIT 区域、由于某些重要的扩展不可用、由于收到了一个位置的错误、由于返回分页结果、由于指出一个过度规定的过滤器、由于返回一个或多个通告属性、或者由于操作结果是一个流结果并且该响应不是结果的最后一个响应，而使结果不完整时，将出现该参数。

a） **LimitProblem** 参数用于指出是否已经超出了时间限制、大小限制或管理限制。返回的结果为当达到限度时可用的那些结果。

b） 如果不开发 DIT 区域，那么 **unexplored** 参数将出现。其信息允许 DUA 通过联系其他访问点（如果它这么选择的话）来继续处理列表操作。参数包括一系列（可能为空）**ContinuationReferences**，每个包括基对象（在其上可以进行操作）的名称、**OperationProgress** 的适当值、一系列访问点（在其上可以进一步进行请求）。返回的 **ContinuationReferences** 将处于在操作服务控制请求的提名范围内。见第 12.6 节。

c） **unavailableCriticalExtensions** 参数指出，如果出现，在号码簿的某些部分，一个或多个重要扩展是不可用的。

d） **unknownErrors** 参数用于返回未知的错误类型或在操作处理中自其他 DSA 接收的参数。SET 的每个成员都包含一个这样的未知错误。见 ITU-T X.519 建议书| ISO/IEC 9594-5 第 12.2.4 节。

e） 当 DUA 已请求分页结果并且 DSA 未返回所有的可用结果时，**queryReference** 参数将出现。见第 7.9 节。当 DSA 能够确定对用户有效的所有结果都已返回时，它将不存在（即，它不是一个应用访问控制的结果）。

f） **overspecFilter** 分量只与搜索操作一起使用，当作为过度规定过滤的结果，返回的搜索结果为空时，虽然存在候选的条目，它们只匹配于部分过滤器，或者只大致匹配于过滤器。只有当搜索请求包括 **checkOverspecified** 项并且号码簿能够确定过滤器过度规定了时，才返回它。它包括在 **search** 变量中提供的过滤器，利用成功匹配的过滤器的那些元素，删去某些条

目。产生 **overspecFilter** 的实际程序是一个局部问题。

注 1 — 分布式号码簿中适当 **overspecFilter** 的返回有待进一步研究。

g） **notification** 参数可以用来发送错误结果限定，并可针对使用的搜索操作，返回一个 **proposedRelaxation** 属性（见 ITU-T X.520 建议书| ISO/IEC 9594-6 第 5.12.15 节，它提供了一种宽松策略，可供用户使用。在这种情况下，可以提供 **MRMapping** 元素序列，它将用于影响宽松（或紧缩）策略，由有关的搜索规则规定。

注 2 — **notification** 中 **sequence-of Attribute** 的次序并不重要。

h） **entryCount** 参数只与 **search** 结果有关，并且如果出现，那么它将对满足搜索准则要求的条目数量给出一个最佳的估计。该子分量将出现，当且仅当：

— 在搜索变量中或者通过管理搜索规则设置了 **entryCount** 搜索控制选项；

— 如果已经请求了分页结果或者超出了大小限制；以及

— 如果至少一个参数 DSA 支持该特性。

当 **entryCount** 子分量出现时，如果所有执行 DSA 都支持该特性，并且如果所有符合要求的 DSA 都参与了操作，那么将采用 **bestEstimate** 或准确选择。如果所有参与的 DSA 都能提供一个准确的计数，那么将采用准确的选择，否则将采用 **bestEstimate** 选择。如果不是所有符合要求的 DSA 都参与了操作，或者部分参与的 DSA 不支持 **entryCount** 参数，那么将采用 **lowEstimate** 选择。复合条目的族成员只当作是一个单个条目。

i） **streamedResult** 参数指明，当出现并为 TRUE 时，DSA 发送一个流结果并且该响应不是结果的最后响应。如果不存在或以 FALSE 出现，该参数指明该响应是流结果的最后响应或者它是一个非流响应。流结果中的每个响应将用相同的 **invokeId** 进行确定。

如果遇到限制问题，它将导致在 **PartialOutcomeQualifier** 中使用 **limitProblem** 元素，那么该分量将在所有作为分页结果集的一部分而提供的所有后续结果中予以重复。

注 3 — 流结果中的每个响应都将利用同一 **invokeID** 进行确定。这样，只有 IDM 号码簿协议可以使用该选项，如 ITU-T X.519 建议书| ISO/IEC 9594-5 所规定。

当 DUA 已经提出了标记保护请求时，或者如果出于其他原因致使号码簿无法关联信息，那么 **uncorrelatedListInfo** 参数可以包含众多源自号码簿不同分量并由之标记的结果参数集。如果在链接中没有任何 DSA 能够关联所有的结果，那么 DUA 将从各种不同的片断中组装实际的结果。

CommonResults（见第 7.4 节）包括适用于响应的安全参数。如果号码簿对结果进行标记、加密或者标记和加密，那么 **SecurityParameters** 分量（见第 7.10 节）将包括在结果中。

10.1.4 列表错误

如果请求失败，那么将报告其中一个列出的错误。对将报告特殊错误的情况在第 12 节中进行定义。

10.1.5 基本访问控制的列表操作决策点

如果 **rule-based-access-control** 也应用，那么有关 **basic-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **basic-access-control** 对正在执行 **list** 操作的那部分 DIB 起作用，那么应用以下访问控制序列：

1） 对由 **object** 变量确定的条目不需要任何特殊的许可。

2） 对每个将在 **subordinates** 中返回一个 **RelativeDistinguishedName** 的直接下属，对该条目需要浏览和 *ReturnDN* 许可。将忽略那些未赋予许可的条目。如果作为应用这些控制的结果，没有返回任何下属信息（**PartialOutcomeQualifier** 中的任何 **ContinuationReferences** 除外），并且如果未将 *DiscloseOnError* 许可赋予给由 **object** 变量确定的条目，那么操作失败，并将返回一个带问题 **noSuchObject** 的 **nameError**。**matched** 元素将包含下一个上级

条目的名称，对该条目赋予了 *DiscloseOnError* 许可，或者将包含 DIT 根的名称（即一个空 **RDNSequence**）。否则，操作成功，但它不传送任何下属信息（**PartialOutcomeQualifier** 中的任何 **ContinuationReferences** 除外）。

注 1 — 在返回 **nameError** 的情况下，未访问所有上级条目的 DSA 可能使用空 **RDNSequence**。

注 2 — 安全策略可以防止泄漏下属信息，否则将作为 **PartialOutcomeQualifier** 中的 **ContinuationReferences** 予以传送。如果这样一个策略发挥作用，并且如果 DUA 通过规定 **chainingProhibited** 来约束服务，那么号码簿可能返回一个带问题 **chainingRrequired** 的 **serviceError**。否则将接着执行上面 2）中所述的程序。

注 3 — 安全策略可以防止号码簿指明一个列出的下属条目是一个别名条目。例如，如果 DUA 未将读访问赋予别名条目、其包含的 **objectClass** 属性和值 **alias**，那么号码簿可能从 **ListResult** 中删去 **subordinates** 的 **aliasEntry** 分量，或者将之设为 **FALSE**。

注 4 — 如果未将 *DiscloseOnError* 许可赋予 **object** 变量确定的条目，那么不应返回指明 **limitProblem** 或 **unavailableCriticalExtensions** 的 **partialOutcomeQualifier**，原因是它可能对本条目的安全造成危害。

10.1.6 基于规则的访问控制的列表操作决策点

如果 **basic-access-control** 也应用，那么有关 **rule-based-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **rule-based-access-control**、**rule-and-basic-access-control** 或 **rule-and-simple-access-control** 对正在执行 **List** 操作的那部分 DIB 起作用，那么应用以下访问控制序列：

1） 如果对由 **object** 变量确定的条目拒绝基于规则的条目级许可，那么将依据第 7.11.2.4 节返回带问题 **noSuchObject** 的 **nameError**。

2） 对每个将在 **subordinates** 中返回 **RelativeDistinguishedName** 的直接下属，务必将基于规则的 RDN 许可赋予给该条目。忽略未准予访问的各条目。

3） 应用 **basic-access-control**，如第 10.1.5 节所述。

10.2 搜索

10.2.1 搜索语法

搜索操作对感兴趣的条目搜索号码簿的一个或多个部分，并从这些条目返回选定的信息。操作变量可以由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

```
search  OPERATION  ::= {
        ARGUMENT        SearchArgument
        RESULT          SearchResult
        ERRORS          { attributeError | nameError | serviceError | referral | abandoned |
                        securityError }
        CODE            id-opcode-search }

SearchArgument  ::=  OPTIONALLY-PROTECTED {
        SET {
                baseObject              [0]     Name,
                subset                  [1]     INTEGER {
                        baseObject(0), oneLevel(1), wholeSubtree(2) } DEFAULT baseObject,
                filter                  [2]     Filter DEFAULT  and : { },
                searchAliases           [3]     BOOLEAN  DEFAULT TRUE,
                selection               [4]     EntryInformationSelection DEFAULT { },
                pagedResults            [5]     PagedResultsRequest OPTIONAL,
                matchedValuesOnly       [6]     BOOLEAN DEFAULT FALSE,
                extendedFilter          [7]     Filter OPTIONAL,
                checkOverspecified      [8]     BOOLEAN DEFAULT FALSE,
                relaxation              [9]     RelaxationPolicy                OPTIONAL,
                extendedArea            [10]    INTEGER                         OPTIONAL,
                hierarchySelections     [11]    HierarchySelections  DEFAULT { self },
                searchControlOptions    [12]    SearchControlOptions        DEFAULT { searchAliases },
                joinArguments           [13]    SEQUENCE SIZE (1..MAX) OF JoinArgument OPTIONAL,
                joinType                [14]    ENUMERATED {
```

```
                innerJoin(0), leftOuterJoin(1), fullOuterJoin(2) } DEFAULT leftOuterJoin,
            COMPONENTS OF       CommonArguments } }

HierarchySelections  ::=  BIT STRING {
        self                (0),
        children            (1),
        parent              (2),
        hierarchy           (3),
        top                 (4),
        subtree             (5),
        siblings            (6),
        siblingChildren     (7),
        siblingSubtree      (8),
        all                 (9) }

SearchControlOptions  ::=  BIT STRING {
        searchAliases                   (0),
        matchedValuesOnly               (1),
        checkOverspecified              (2),
        performExactly                  (3),
        includeAllAreas                 (4),
        noSystemRelaxation              (5),
        dnAttribute                     (6),
        matchOnResidualName             (7),
        entryCount                      (8),
        useSubset                       (9),
        separateFamilyMembers           (10),
        searchFamily                    (11) }

JoinArgument  ::=  SEQUENCE {
        joinBaseObject          [0]     Name,
        domainLocalID           [1]     DomainLocalID OPTIONAL,
        joinSubset              [2]     ENUMERATED {
                    baseObject(0), oneLevel(1), wholeSubtree(2) } DEFAULT baseObject,
        joinFilter              [3]     Filter  OPTIONAL,
        joinAttributes          [4]     SEQUENCE SIZE (1..MAX) OF JoinAttPair OPTIONAL,
        joinSelection           [5]     EntryInformationSel0ection }

DomainLocalID       ::= DirectoryString { ub-domainLocalID }
```

DomainLocalID 是一个字符串，在本地唯一确定一个部分持有另一个 DIT 的远程域。

注 — 该字符串在本地定义，无需由任何注册权威部门进行注册。

```
JoinAttPair  ::=  SEQUENCE {
        baseAtt                 AttributeType,
        joinAtt                 AttributeType,
        joinContext             SEQUENCE SIZE (1..MAX) OF JoinContextType OPTIONAL }

JoinContextType  ::=  CONTEXT.&id({SupportedContexts})

SearchResult  ::=  OPTIONALLY-PROTECTED {
        CHOICE {
            SearchInfo                  SET {
                    name                            Name  OPTIONAL,
                    entries                 [0]     SET OF EntryInformation,
                    partialOutcomeQualifier [2]     PartialOutcomeQualifier OPTIONAL,
                    altMatching             [3]     BOOLEAN  DEFAULT FALSE,
                    COMPONENTS OF                   CommonResults },
            uncorrelatedSearchInfo          [0]     SET OF SearchResult } }
```

10.2.2 搜索变量

baseObject 变量用于确定相对将要进行的主要搜索的对象条目（或可能的话为根）。**baseObject** 可以是一个可选的名称，并包括正文信息，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 9.3 节所述。

subset 变量指明主要搜索是否应用于：

a） 只有 **baseObject**；

b） 只有基对象的直接下属（**oneLevel**）；

c） 基对象及其所有下属（**wholeSubtree**）。

如果基对象是一个普通条目，那么依据 **subset** 规定，认为复合条目是单个条目。如果基对象是复合条目的祖先，那么 **searchFamily** 搜索控制选项将控制准确的行为。如果基对象是一个子族成员，那么认为族成员是单个条目。

filter 变量用于从不感兴趣的主要搜索空间去除条目。只在符合过滤器要求的条目上返回信息（见第 7.8 节）。在出现基本的用户提供的或搜索规则提供的宽松策略情况下，将以要求的匹配规则替换，在第一时间对过滤器进行评估。

在出现用户提供的或搜索规则提供的宽松策略情况下，或者在二者都出现的情况下，如果返回的结果比最低的要求还要少，那么将对过滤器重新进行评估，利用适当的宽松策略（见第 7.8 节和以下内容，它们有关 **SearchArgument** 的宽松元素），逐步递增，直至有足够多的条目或者没有更多的宽松策略可定义。同样，如果返回的结果比最高的要求还要多，那么也将对过滤器重新进行评估，利用适当的宽松策略，逐步递增，直至有足够少的条目或者没有更多的紧缩策略可定义。

注 1 — 如果未提供搜索规则的放宽条件，那么用户可能需要对过滤器进行简化，并做再次努力，或者可选地定义一个用户定义的放宽条件。

CommonArguments 的 **familyGrouping** 分量用于在应用过滤器之前，在逻辑上将各条目合并进一个族中，如第 7.3.2 节和第 7.8.3 节所述。

当找到基对象后，依据 **dontDereferenceAliases** 服务控制设置，各别名将被废弃。基对象各下属中的别名将依据 **searchAliases** 参数设置，在搜索期间被废弃。如果 **searchAliases** 参数为 **TRUE**，那么各别名将被废弃，如果参数为 **FALSE**，那么各别名将不被废弃。如果 **searchAliases** 参数为 **TRUE**，那么将在别名条目的子树中继续进行搜索。

selection 变量指明自条目请求什么信息（见第 7.6 节）。不过，不应假设返回的属性等同于请求的那些属性或限于请求的那些属性。

注 2 — 出于分布式操作的目的，用于协调相关条目分布式操作的 DSA（即已完成对包含 **joinArguments** 的搜索变量的名称解析，并需要从非内部来源中获得一个潜在相关的条目集）需要覆盖 DAP 提供的、带 **attributeTypesAndValues** 的 **infoTypes**，并且在选择需要利用分布式操作返回的属性中，还需要包括连接属性（即由 **JoinArgument.joinAttributes** 中 **JoinAttPair.joinAtt** 规定的集合中的属性）。不过，如果 **infoTypes** 值为 **attributeTypesOnly**，那么通过协调 DSA 返回给用户的条目和派生的条目将删去 DAP 返回信息中的属性值，并将因此依据最初的用户请求返回 **EntryInformation**。

pagedResults 变量用于请求应逐页返回操作结果，如第 7.9 节所述。

matchedValuesOnly 变量用于指出将从返回的条目信息中删去某些属性值。尤其是，当返回的属性是多值的，并且某些但不是所有的属性值都对搜索过滤器起作用，以其最后的有效形式（即考虑宽松的匹配规则）通过 **present** 之外的过滤器项返回 TRUE，那么从返回的条目信息中删去不那么起作用的值。

如果在 **search** 变量中规定了 **matchedValuesOnly** 变量，那么对将要返回的属性应用以下逻辑处理过程：

a） 如果过滤器由一个过滤器项组成，那么应用以下规则：

— 如果过滤器项类型为 **present**，那么 **matchedValuesOnly** 变量对该过滤器项中的属性不起作用。

— 如果过滤器项类型为 **equality**、**substrings**、**greaterOrEqual**、**lessOrEqual**、**approximateMatch**、**contextPresent** 或 **extensibleMatch**，并且对属性的命题不为 TRUE，那么 **matchedValuesOnly** 变量对该属性不起作用。如果命题为 TRUE，那么将从返回的条目信息中删去不匹配过滤器项的该属性的值。

— 如果对过滤器项求反，那么 **matchedValuesOnly** 变量对该属性不起作用。

b） 如果过滤器是复杂的（包括多个过滤器项），那么应用以下规则：

— 如果过滤器包含一个求反（即 **not**）过滤器，那么 **matchedValuesOnly** 变量对求反过滤器中的任何属性都不起作用。

注 3 — 这还适用于嵌套的、求反的过滤器。

— **matchedValuesOnly** 变量对或（即 **or**）过滤器的任何元素的属性都不起作用，评估为 FALSE 或 UNDEFINED。

— 对一个在过滤器中出现多次的属性只需其中一次出现评估为 TRUE，如上面 a）第 2 点所述，对有效的 **matchedValuesOnly** 变量，即一次有效将覆盖一次或多次忽略。

— 对 **or** 过滤器中的每个过滤器都应评估 **matchedValuesOnly**，即使过滤器的真值可以在彻底评估完成之前确定。

在混合版本情况下，使用 **extendedFilter** 变量来规定上述中的一个可选过滤器。当该变量出现时，**filter** 变量（如果有的话）将被第二版本和后续版本的系统所忽略。**extendedFilter** 总被第一版本系统所忽略。搜索宽松策略仅用于 **filter**。

注 4 — 通过包括两个过滤器，在搜索请求的分布式处理中，DUA 就可以规定第一版本系统使用一个过滤器，第二版本和后续版本系统使用另一个不同的过滤器。第一版本系统不支持属性多态性或匹配规则命题。

如果搜索操作的结果为空并且号码簿能够确定这是因过滤器过度规定而造成的，那么将用 **checkOverspecified** 变量来请求号码簿返回一个 **partialOutcomeQualifier** 中的 **overspecFilter** 项。

可以用 **relaxation** 分量来规定一个用户提供的 **RelaxationPolicy**，利用 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 16.10 节中定义的结构。

依据管理搜索规则，如果替换将引起搜索无效，那么 **search** 请求规定的替换将不在服务特定管理区域内执行。当替换匹配规则出现以下情况时，将与管理搜索规则发生冲突：

a） 从 **search** 过滤器中有效地移去一个或多个过滤器项；或者

b）与属性类型的 **matchingUse** 规定发生冲突（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 16.10.2 节）。

注 5 — **nullMatch** 匹配规则可以将一个或多个过滤器项从过滤器中移去。当使用该匹配规则时，管理搜索规则可能冲突。

如果在服务特定管理区域外执行搜索操作，或者如果管理搜索规则未提供 **RelaxationPolicy** 分量，那么应用用户提供的 **RelaxationPolicy**，如 ITU-T X.501 建议书| ISO/IEC 9594-2 第 16.10.7 节所述。当搜索规则提供的 **RelaxationPolicy** 也出现时，依据以下程序，实施结合：

1） 搜索规则规定的基本替换策略，如果有的话，在搜索确认过程中应用。因此而进行管理搜索规则规定的、可能的基本替换。

2） 在 **search** 请求中规定的基本替换和基于映射的映射，如果出现的话，将应用。不过，不应用将引起管理搜索规则冲突的基本替换，而是忽略之。在这种情况下，**oldMatchingRule** 值（如果提供了的话）适用于基本的匹配规则，即在搜索规则应用的基本替换策略不存在的情况下，它将适用。

3） 宽松/紧缩替换，如果有的话，在 **search** 请求中规定，而后与任何规定的、基于映射的匹配一起使用，依据的是在 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 16.10.7 节中规定的规则。如果在任何引起与管理搜索规则不一致的点遇到替换匹配规则，那么彻底放弃该特殊替换，以及任何由 **search** 请求为该属性类型规定的进一步的替换。如果在该过程中，**search** 请求中规定的 **minimum** 或 **maximum** 规定得到了满足，那么停止该过程。

4） 应用管理搜索规则提供的宽松或紧缩替换，例外是，对已执行了宽松或紧缩替换的属性类型，不进行任何替换。也就是说，进一步的宽松或紧缩替换只适用于到目前为止尚未进行宽松或紧缩替换的属性类型的匹配规则。在这部分过程中，将继续使用 **search** 请求中的 **maximum** 或 **minimum** 规定，而不使用那些在管理搜索规则中规定的规定。

如果在 **search** 请求中规定的替换提议了一个不支持的匹配规则，那么现有的匹配规则将继续发挥作用。如果该策略无法产生一个支持的匹配规则，那么过滤器项被评估为 UNDEFINED。

用户可以提议系统通过规定哑匹配规则 **systemProposedMatch** 来提供某种宽松或紧缩。

extendedArea 分量用于指明宽松的程度（如果大于 0 的话）或者紧缩的程度（如果小于 0 的话）。如果该分量出现，那么它对宽松或紧缩有影响，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 16.10.7 节所述。

hierarchySelection 搜索控制通过一个比特串来规定将要在每个匹配条目层次型组中执行的层次型选择。对不是层次型组一部分的匹配条目将忽略之。如果匹配一个层次中的若干条目，那么层次型选择将不产生返回多次的相同条目。如果该搜索控制不出现，那么不执行任何层次型选择。当出现时，以下选择可能是单独的或结合的：

a） **self** 指明，应从匹配条目返回条目信息。如果这是唯一选择，那么它对应的是不进行任何层次型选择。

b） **children** 指明，对每个匹配的条目，如果有的话，从每个匹配条目的所有直接层次下属返回条目信息。如果这是唯一设置，那么没有任何信息从匹配条目返回。

c） **parent** 指明，对每个匹配的条目，如果有的话，从每个匹配条目的所有直接层次上级返回条目信息。如果这是唯一设置，那么没有任何信息从匹配条目返回。

d） **hierarchy** 指明，对每个匹配的条目，从所有层次上级返回条目信息。如果这是唯一设置，那么没有任何信息从匹配条目返回。

e） **top** 指明，对每个匹配的条目，从层次顶层返回条目信息。如果这是唯一设置，那么没有任何信息从匹配条目返回，除非匹配条目是顶层条目。

f） **subtree** 指明，对每个匹配的条目，如果有的话，从其所有层次下属返回条目信息。如果这是唯一设置，那么没有任何信息从匹配条目返回。

g） **siblings** 指明，对每个匹配的条目，从所有层次同胞返回条目信息。如果这是唯一设置，那么没有任何信息从匹配条目返回。

h） **siblingChildren** 指明，对每个匹配的条目，从所有层次同胞的直接层次下属返回条目信息。如果这是唯一设置，那么没有任何信息从匹配条目及其同胞返回。

i） **siblingSubtree** 指明，对每个匹配的条目，从所有层次同胞的所有下属返回条目信息。如果这是唯一设置，那么没有任何信息从匹配条目及其同胞返回。

j） **all** 指明，对每个匹配的条目，从层次族的所有条目返回条目信息。

searchControlOptions 分量只包含适用于搜索操作的控制选项。该分量拥有语义等同于搜索变量布尔类型分量语义的指示器。一个支持服务管理扩展的实现方案将支持该分量。一个发送支持的实现方案（例如一个 DUA）在设置布尔类型分量之外还将设置该分量的各对应位（除非应用缺省值）。如果一个支持 DSA 的实现方案用该分量接收一个 **search** 请求，那么它将忽略请求中的布尔类型分量。如果在请求中不存在该分量，那么缺省设置将被理解为重新设置所有位，除非如下所述：

a） **searchAliases** 搜索控制选项是对 **searchAliases** 搜索变量分量的替换。如果设置了该位，那么它对应值为 **TRUE** 的 **searchAliases** 分量。如果 **searchControlOptions** 分量不存在，那么缺省值取决于 **searchAliases** 分量，即如果 **searchAliases** 分量不存在或者设为 **TRUE**，那么该位缺省为设置值。

b） **matchedValuesOnly** 搜索控制选项是对 **matchedValuesOnly** 搜索变量分量的替换。如果设置了该位，那么它对应值为 **TRUE** 的 **matchedValuesOnly** 分量。如果 **searchControlOptions** 分量不存在，那么缺省值取决于 **matchedValuesOnly** 分量，即如果 **matchedValuesOnly** 设为 **TRUE**，那么该位缺省为设置值；否则该位缺省为重新设置。

c） **checkOverspecified** 搜索控制选项是对 **checkOverspecified** 搜索变量分量的替换。如果设置了该位，那么它对应值为 **TRUE** 的 **checkOverspecified** 分量。如果 **searchControlOptions** 分量不存在，那么缺省值取决于 **checkOverspecified** 分量，即如果 **checkOverspecified** 分量设为 **TRUE**，那么该位缺省为设置值；否则该位缺省为重新设置。

d） **performExactly** 搜索控制选项指明，合适的话，在替换基本的匹配规则后，将严格按照过滤器规定的或暗指的相关匹配规则，执行一个操作。当 **extensibleMatch** 过滤器项规定了一个不支持的匹配规则时，如果设置了该搜索控制选项，那么将拒绝 **search** 请求。否则，过滤器项评估为 UNDEFINED。如果搜索操作在服务特定管理区域内开始其初始评估阶段，并且搜索规则中的匹配限制出现冲突，那么当且仅当设置了该搜索控制选项，搜索规则将使搜索确认失败。

e） 只有当 **extendedArea** 分量包括一个 0 或更大值时，**includeAllAreas** 搜索控制选项才相关。在所有其他情况下，它将被忽略。如果值为 **TRUE**，那么执行包含的宽松；否则如果可能的话，执行排他的宽松（见 ITU-T X.501 建议书| ISO/IEC 9594-2 第 13.6 节）。

f） 当用户要求不使用 DSA 提供的宽松策略时，使用 **noSystemRelaxation** 搜索控制选项。DSA 仍将使用基本策略，除非有一个覆盖它的用户提供的基本策略，但将不能使用任何后续的宽松或紧缩策略。也就是说，对候选条目集，过滤器从不评估一次以上，除非由于用户提供的宽松策略。

g） **dnAttribute** 搜索控制选项用于指明，除了当依据条目对过滤器进行评估时所用的那些条目属性外，还使用了哪些条目的不同名称的属性。如果设置，它将覆盖 **extensibleMatch** 过滤器项中任何可能的 **dnAttribute** 规定。它还适用于所有的过滤器项类型。

h）只有当设置了 **partialNameResolution** 搜索控制选项时，**matchOnResidualName** 搜索控制选项才相关。它用于指明，如果号码簿只能解析 **search** 操作中的部分假设名称，那么未解析 RDN 的 AVA 将被当作是经过 AND（与）运算的 **equality** 过滤器项。这些过滤器项与搜索过滤器做 AND（“与”）运算，针对的是依据搜索规则的搜索评估和条目匹配。

i） **entryCount** 搜索控制选项指明，在超出了服务控制大小限制或管理大小限制的情况下，将在 **search** 结果中提供一个条目计数。**entryCount** 指明，返回多少条目将拥有一个未遇到过的大小限制。如果设置了 **subentries** 服务控制选项，那么将忽略该搜索控制。

j） **useSubset** 搜索控制选项指明，将忽略 **imposedSubset** 搜索规则分量（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 16.10.9 节）。

k） **separateFamilyMembers** 搜索控制选项指明，族成员将作为单独的条目而非嵌入在 **family-information** 派生属性中返回。

l） 如果基对象为复合属性的一个祖先，那么 **searchFamily** 搜索控制选项将规定如何执行搜索。如果基对象不是一个祖先，或者如果在 **CommonArguments** 或 **ChainingArguments** 中设置了 **entryOnly**，那么忽略该选项。如果设置了该选项，那么依据 **subset** 和 **sizeLimit** 规定，只对复合条目执行该操作，并将每个族成员当作一个单独的条目。如果未设置 **searchFamily** 选项，那么依据 **subset** 规定，认为复合条目是一个单个条目。

注 6 — 后者意味着，作为例子，如果 **subset** 设为 **baseObject**，并且 **familyGrouping** 为 **entryOnly**，那么每个单个族成员都在搜索范围内。

JoinArguments 变量用于规定号码簿的额外部分，将出于以下目的对其进行搜索，即确定和访问与主要搜索相关的条目，并规定将在连接相关条目中使用的属性。虽然规定为一个 SEQUENCE，但 **joinArgument** 变量出现的次序并不重要。

注 7 — 当规定 **joinArguments** 时，认为主要搜索和每个额外搜索都将产生一系列中间结果。来自 **joinArgument** 规定的每个中间结果集将与主要搜索的结果相连接，所有的连接都将在返回 **SearchResult** 中的任何结果之前执行。各中间结果对号码簿用户是不可见的。

joinBaseObject 变量用于确定相对每个将要进行的额外搜索的对象条目（或可能的话为根）。**joinBaseObject** 可以是一个可选的名称，并包括正文信息，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 9.3 节所述。

domainLocalID 变量用于任选地确定一个独立的 DIT，在其中将启动对 **joinBaseObject** 的搜索。如果不存在，那么对 **joinBaseObject** 的搜索将在 DSA 所知的所有 DIT 中启动。

joinSubset 变量指明额外搜索是否应用于：

a） 只有 **joinBaseObject**；

b） 只有连接基对象的直接下属（**oneLevel**）；

c） 连接基对象及其所有下属（**wholeSubtree**）。

joinFilter 变量用于从不感兴趣的额外搜索空间去除条目。对连接相关条目，将只考虑符合 **joinFilter** 要求的信息。如果不规定 **joinFilter**，那么将使用 **SearchArgument filter** 分量中的值。如果未提供 **SearchArgument** 的 **filter** 分量，那么将使用该分量的缺省值。当出现时，依据有关 **extendedFilter** 的规则，将对 **joinFilter** 进行处理。

joinAttributes 变量用于规定各属性对，它们将用于连接来自主要搜索的条目和来自额外搜索的条目。如果存在一个 **joinAttrPair**，使以下条件为 TRUE，那么认为一个来自主要搜索的条目（“主要条目”）与一个来自额外搜索的条目（“额外条目”）相关：

a） 主要条目拥有一个由 **baseAtt** 为属性类型规定的值；

b） 额外条目拥有一个由 **joinAtt** 为属性类型规定的值；

c） 依据以下规则，主要条目中的一个属性值和额外条目中的一个属性值是相同的：

i） 如果属性类型相同，那么对该属性类型使用等同的匹配规则；

ii） 如果属性类型不相同，但有相同的语法，那么对为主要条目规定的属性类型使用等同的匹配规则；

iii） 如果 **joinContexts** 出现，那么依据上面规则 i）或 ii），在评估中只能使用规定正文的属性值。如果 **joinContexts** 不存在，那么依据上面规则 i）或 ii），在评估中可以使用所有正文的属性值。

在为潜在的连接评估 **joinAttributes** 中，将忽略连接属性的子类型。只有明确确定的 **baseAtt** 和 **joinAtt** 才会用于评估一个潜在的连接。

如果应用一个等同规则，并评估为 FALSE 或 UNDEFINED，那么不认为各条目是相关的。

如果在上述条件 c）下没有合适的匹配规则可用，那么不认为各条目是相关的。

注 8 — 当规定涉及多值属性的连接时，应注意防止无意地搜索没有意义的数据。例如，如果条目使用一个多值属性，如雇员标识符，来表示委员会中的成员资格，那么在执行连接中该多值属性的规定将返回一个包含族成员名称、电话号码、电子邮件等的无关联集。不过，当规定外部连接时，将返回所有的被检索条目，即使它不相关。

joinSelection 变量用于从不感兴趣的额外搜索中间结果中去除属性。

joinType 变量用于规定将对相关条目执行的连接类型，如下所述：

a） 如果规定了 **innerJoin**，那么结果条目集将只包括那些执行了连接的条目，它基于 **joinAttributes** 中规定的属性对。每个结果条目都将包括所有对应的相关条目，作为 **relatedEntry** 属性值。

b） 如果规定了 **leftOuterJoin**，那么结果条目集将包括所有由主要搜索选择的条目；所有执行了连接的条目（基于 **joinAttributes** 中规定的属性对）都将包括所有对应的相关条目，作为 **relatedEntry** 属性值。

c） 如果规定了 **fullOuterJoin**，那么结果条目集将包括所有来自主要搜索和额外搜索的条目；所有执行了连接的条目（基于 **joinAttributes** 中规定的属性对）都将包括所有对应的相关条目，作为 **relatedEntry** 属性值，而不是作为明确的条目。

除非 **joinAttributes** 值包含至少一个 **JoinAttPair**，并且依据匹配规则，每个 **JoinAttPair** 都是有效的，否则不得尝试任何连接。如果不是这种情况，那么不得尝试任何连接，并且将以下作为合并各 **JoinAttPair** 的结果，依据的是连接类型：

连接类型	合并后的输出
inner-join	空
left-outer-join	只有主要结果
full-outer-join	来自主要搜索和连接搜索的结果

否则，只有当提供所有的相关连接属性值时，条目才适于连接。

连接结果将包括匹配的连接属性的所有组合。

注 9 — 例如，考虑 A、B、C（作为来自主要搜索的条目）、P、Q、R（作为来自使用 J 的额外搜索的条目）、对应的 **JoinAttPair** 值，并假设发生以下匹配是 J 的结果：

— A 与 P、A 与 Q、A 与 R

— B 与 Q

— C 与 P 以及 C 与 Q

而后连接的结果将包括：

— A 与{P，Q，R}

— B 与{Q}

— C 与{P，Q}

即使 Q 的结果出现三次。

CommonArguments（见第 7.3 节）包括适用于请求的服务控制规定和安全参数。如果请求方对该操作的变量进行标记、加密或者标记和加密，那么将在变量中包括 **SecurityParameters**（见第 7.10 节）分量。

10.2.3 搜索结果

依据访问控制，如果找到了 **baseObject**，而不管是否返回下属，并且如果在服务特定的管理区域内没有规定任何阻止搜索操作继续进行的服务限制，那么请求成功。

注 1 — 作为其必然结果，对查询同一条目属性集的读操作而言，适用于单个条目的未过滤搜索的结果可以不相同。这是因为如果在条目中不存在任何选定的属性，那么后者将返回一个 **AttributeError**。

name 为条目的不同名称或条目的别名，如第 7.7 节所述。只有当别名废弃时、当 RDN 已解析为主要 RDN 时，或者当内容选择已应用时、当将要返回的名称不同于操作变量中所提供的 **baseObject** 名称时，它才出现。

entries 参数从各个（0 个或多个）满足过滤器要求的条目传送请求的信息（见第 7.5 节）。作为 **entries** 一部分提供的名称可能会受第 7.7 节中为 **Name** 所述的正文影响。条目信息可以包括如 **EntryInformationSelection familyReturn** 元素要求的族信息。**familyGrouping** 与 **familyReturn** 之间的交互作用在过滤器的四阶段评估中以及返回内容的后续评估中进行定义，如第 7.8.3 节所述。

partialOutcomeQualifier 如第 10.1.3 节所述。

注 2 — 如果某个特定条目的返回条目信息不完整，那么它通过返回条目信息中的 **incompleteEntry** 参数来指出。

altMatching 用于指明未按 **search** 请求中规定的要求准确应用匹配规则。

CommonResults notifications 元素中的 **appliedRelaxation** 属性用于列出已放宽或收紧的过滤器属性，而不是放宽策略 **basic** 元素提出的那些属性（见 ITU-T X.520 建议书 | ISO/IEC 9594-6 第 5.12.16 节）。

所描述的 **uncorrelatedSearchInfo** 参数针对的是第 10.1.3 节中的 **uncorrelatedListInfo**。

CommonResults（见第 7.4 节）包括适用于响应的安全参数。如果号码簿对该结果进行标记、加密或者标记和加密，那么将在结果包括 **SecurityParameters**（见第 7.10 节）分量。

10.2.4 服务管理

管理权威部门可以建立服务特定的管理区域，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 7 节所述。这使得管理权威部门能够通过限制搜索操作来对服务进行管理，它通过定义搜索规则来限定可以搜索的 DIT 区域、可以形成的搜索类型、可以返回的信息等。

10.2.5 搜索错误

如果请求失败，那么将报告其中一个列出的错误。对将报告特殊错误的情况在第 12 节中进行定义。

当在服务特定的管理区域内执行搜索时，可以返回众多额外的、非常详细的错误信息元素，详细内容见第 13 节。

10.2.6 基本访问控制的搜索操作决策点

如果 **rule-based-access-control** 也应用，那么有关 **basic-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **basic-access-control** 对将要搜索的那部分 DIT 起作用，那么应用以下访问控制序列：

1） 对由 **baseObject** 变量确定的条目不需要任何特殊的许可。

注 1 — 如果 **baseObject** 处于 **SearchArgument** 范围内（即当 **subset** 变量规定 **baseObject** 或 **wholeSubtree** 时），那么应用 2）-5）中规定的访问控制。

2） 对在 **SearchArgument** 范围内的每个条目（将作为候选考虑对象），需要浏览许可。将忽略未赋予该许可的各条目。

3） **filter** 变量适用于每个留待考虑 2）后才考虑的条目，依据的是以下内容：

a） 对每个规定一个属性的 **FilterItem**，在 **FilterItem** 被评估为 TRUE 或 FALSE 之前，对属性类型需要 *FilterMatch* 许可。未赋予该许可的 **FilterItem** 评估为 UNDEFINED。

b） 对每个额外规定一个属性值的 **FilterItem**，对每个保存的属性值（将考虑把它用于匹配目的）都需要 *FilterMatch* 许可。如果存在一个匹配 **FilterItem** 的值，并且赋予了许可，那么 **FilterItem** 评估为 TRUE，否则评估为 FALSE。

4） 如果出现，那么 **joinCriteria** 变量适用于每个留待考虑 3）后才考虑的条目，依据的是以下内容：

a） 对每个规定一个属性的 **JoinCriterialtem**，在 **JoinCriterialtem** 被评估为 **TRUE** 或 FALSE 之前，对属性类型需要 *FilterMatch* 许可。未赋予该许可的 **JoinCriterialtem** 评估为 UNDEFINED。

b） 对每个额外规定一个属性值的 **JoinCriterialtem**，对每个保存的属性值（将考虑把它用于匹配目的）都需要 *FilterMatch* 许可。如果存在一个匹配 **JoinCriterialtem** 的值，并且赋予了许可，那么 **JoinCriterialtem** 评估为 TRUE，否则评估为 FALSE。

5） 一旦应用了在 2）- 4）中定义的程序，那么要么选择条目，要么抛弃条目。如果作为对整个范围内的子树应用这些控制的结果，没有选择任何条目（**partialOutcomeQualifier** 中的任何 **ContinuationReferences** 除外），并且如果未将 *DiscloseOnError* 许可赋予给由 **baseObject** 变量确定的条目，那么操作失败，并将返回一个带问题 **noSuchObject** 的 **nameError**。**matched** 元素将包含下一个上级条目的名称，对该条目赋予了 *DiscloseOnError* 许可，或者将包含 DIT 根的名称（即一个空 **RDNSequence**）。否则，操作成功，但它不传送任何下属信息。

注 2 — 在返回 **nameError** 的情况下，不访问所有上级条目的 DSA 可以使用空 **RDNSequence**。

注 3 — 安全策略可以防止透露知识信息，否则将作为 **partialOutcomeQualifier** 中的 **ContinuationReferences** 予以传送。如果这样一个策略发挥作用，并且如果 DUA 通过规定 **chainingProhibited** 来约束服务，那么号码簿可能返回一个带问题 **chainingRequired** 的 **serviceError**。否则，将从 **partialOutcomeQualifier** 中省略 **ContinuationReference**。

6） 否则，对每个选定的条目，返回的信息如下所述：

a） 如果 **selection** 的 **infoTypes** 元素规定只能返回属性类型，那么对每个将要返回的属性类型，需要读许可。如果未赋予许可，那么从 **EntryInformation** 中删去属性类型。如果作为应用这些控制的结果，没有选择任何属性类型信息，那么返回 **EntryInformation** 元素，但不用它传送任何属性类型信息（即省略 **SET OF CHOICE** 元素或为空）。

b） 如果 **selection** 的 **infoTypes** 元素规定返回属性类型和值，那么对每个将要返回的属性类型和值，需要读许可。如果对属性类型未赋予许可，那么从 **EntryInformation** 中删去属性。如果对属性值未赋予许可，那么从其对应的属性中删去值。在未将许可赋予属性内任何值的情况下，返回一个包含空 **SET OF AttributeValue** 的 **Attribute** 元素。如果作为应用这些控制的结果，没有选择任何属性信息，那么返回 **EntryInformation** 元素，但不用它传送任何属性信息（即省略 **SET OF CHOICE** 元素或为空）。

注 4 — 如果 *DiscloseOnError* 许可未赋予 **baseObject** 变量确定的条目，那么不应返回指出 **limitProblem** 或 **unavailableCriticalExtensions** 的 **partialOutcomeQualifier**，原因是它可能对该条目的安全造成危害。

10.2.6.1 在额外搜索情况下基本访问控制的搜索操作决策点

如果 **joinArguments** 变量出现，并且如果 **basic-access-control** 对将要搜索的那部分 DIT 起作用，那么对每个额外的搜索应用以下访问控制序列：

1） 对由 **joinBaseObject** 变量确定的条目不需要任何特殊的许可。

注 1 — 如果 **joinBaseObject** 处于 **joinArgument** 范围内（即当 **joinSubset** 变量规定 **baseObject** 或 **wholeSubtree** 时），那么应用 2）- 6）中规定的访问控制。

2） 对在 **joinArgument** 范围内的每个条目（将作为候选考虑对象），需要浏览许可。将忽略未赋予该许可的各条目。

3） 如果出现，那么 **joinFilter** 变量适用于每个留待考虑 2）后才考虑的条目，依据的是以下内容：

a） 对每个规定一个属性的 **FilterItem**，在 **FilterItem** 被评估为 TRUE 或 FALSE 之前，对属性类型需要 *FilterMatch* 许可。未赋予该许可的 **FilterItem** 评估为 UNDEFINED。

b） 对每个额外规定一个属性值的 **FilterItem**，对每个保存的属性值（将考虑把它用于匹配目的）都需要 *FilterMatch* 许可。如果存在一个匹配 **FilterItem** 的值，并且赋予了许可，那么 **FilterItem** 评估为 TRUE，否则评估为 FALSE。

4） 如果 **joinFilter** 变量不出现，那么 **filter** 变量适用于每个留待考虑 2）后才考虑的条目，依据的是以下内容：

a） 对每个规定一个属性的 **FilterItem**，在 **FilterItem** 被评估为 TRUE 或 FALSE 之前，对属性类型需要 *FilterMatch* 许可。未赋予该许可的 **FilterItem** 评估为 UNDEFINED。

b） 对每个额外规定一个属性值的 **FilterItem**，对每个保存的属性值（将考虑把它用于匹配目的）都需要 *FilterMatch* 许可。如果存在一个匹配 **FilterItem** 的值，并且赋予了许可，那么 **FilterItem** 评估为 TRUE，否则评估为 FALSE。

5） 一旦应用了在 2）- 4）中定义的程序，那么要么选择条目，要么抛弃条目。如果作为对整个范围内的子树应用这些控制的结果，没有选择任何条目（**partialOutcomeQualifier** 中的任何 **ContinuationReferences** 除外），并且如果未将 *DiscloseOnError* 许可赋予给由 **baseObject** 变量确定的条目，那么操作失败，并将返回一个带问题 **noSuchObject** 的 **nameError**。**matched** 元素将包含下一个上级条目的名称，对该条目赋予了 *DiscloseOnError* 许可，或者将包含 DIT 根的名称（即一个空 **RDNSequence**）。否则，操作成功，但它不传送任何下属信息。

注 2 — 在返回 **nameError** 的情况下，不访问所有上级条目的 DSA 可以使用空 **RDNSequence**。

注 3 — 安全策略可以防止透露知识信息，否则将作为 **partialOutcomeQualifier** 中的 **ContinuationReferences** 予以传送。如果这样一个策略发挥作用，并且如果 DUA 通过规定 **chainingProhibited** 来约束服务，那么号码簿可能返回一个带问题 **chainingRequired** 的 **serviceError**。否则，将从 **partialOutcomeQualifier** 中省略 **ContinuationReference**。

6） 否则，对每个选定的条目，返回的信息如下所述：

a） 如果 **selection** 的 **infoTypes** 元素规定只能返回属性类型，那么对每个将要返回的属性类型，需要读许可。如果未赋予许可，那么从 **EntryInformation** 中删去属性类型。如果作为应用这些控制的结果，没有选择任何属性类型信息，那么返回 **EntryInformation** 元素，但不用它传送任何属性类型信息（即省略 **SET OF CHOICE** 元素或为空）。

b） 如果 **selection** 的 **infoTypes** 元素规定返回属性类型和值，那么对每个将要返回的属性类型和值，需要读许可。如果对属性类型未赋予许可，那么从 **EntryInformation** 中删去属性。如果对属性值未赋予许可，那么从其对应的属性中删去值。在未将许可赋予属性内任何值的情况下，返回一个包含空 **SET OF AttributeValue** 的 **Attribute** 元素。如果作为应用这些控制的结果，没有选择任何属性信息，那么返回 **EntryInformation** 元素，但不用它传送任何属性信息（即省略 **SET OF CHOICE** 元素或为空）。

注 4 — 如果 *DiscloseOnError* 许可未赋予 **baseObject** 变量确定的条目，那么不应返回指出 **limitProblem** 或 **unavailableCriticalExtensions** 的 **partialOutcomeQualifier**，原因是它可能对该条目的安全造成危害。

10.2.6.2 搜索期间废弃别名

对 **search** 操作过程中发生的别名废弃不需要任何特殊的许可（由于 **searchAliases** 参数设为 **TRUE**）。不过，对每个遇到的别名条目，如果别名废弃将导致在 **partialOutcomeQualifier** 中返回 **ContinuationReference**，那么将应用以下访问控制：对别名条目、**aliasedEntryName** 属性及其包含的单个值需要读许可。如果未赋予任何这些许可，那么将从 **partialOutcomeQualifier** 中删去 **ContinuationReference**。这些访问控制也适用于在另一个 DSA 响应中收到的 **continuationReference**。也就是说，DSA 将管辖所有的 **continuationReferences**，不论它们是否在本地产生。

注 — 除了上面所述的访问控制，安全策略可以防止透露知识信息，否则将作为 **partialOutcomeQualifier** 中的 **ContinuationReferences** 予以传送。如果这样一个策略发挥作用，并且如果 DUA 通过规定 **chainingProhibited** 来约束服务，那么号码簿可能返回一个带问题 **chainingRequired** 的 **serviceError**。否则，将从 **partialOutcomeQualifier** 中省略 **ContinuationReference**。

10.2.6.3 不透露不完整的结果

如果在 **EntryInformation** 中返回一个不完整的结果，即因适用的访问控制而删去了某些属性或属性值，那么 **incompleteEntry** 元素将被设为 **TRUE**，条件是 *DiscloseOnError* 许可赋给至少一个限制在结果外的属性类型，或者赋给至少一个限制在结果外的属性值（对该属性类型赋予了读许可）。

10.2.7 基于规则的访问控制的搜索操作决策点

如果 **basic-access-control** 也应用，那么有关 **rule-based-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **rule-based-access-control**、**rule-and-basic-access-control** 或 **rule-and-simple-access-control** 对正在执行 **search** 操作的那部分 DIB 起作用，那么应用以下访问控制序列：

1） 如果对由 **baseObject** 变量确定的条目拒绝基于规则的条目级许可，那么返回带问题 **noSuchObject** 的 **nameError**，如第 7.11.2.4 节定义。

2） 在 **rule-based-access-control** 下，将忽略 **SearchArgument** 范围内的每个条目，对 **SearchArgument**，拒绝条目级访问。

3） 应用有关条目的 **basic-access-control**，如第 10.2.6 节 2）定义。

4） 应用 **filter**，忽略在 **rule-based-access-control** 下拒绝访问的属性值。

5） 应用有关 **filter** 的 **basic-access-control**，如第 10.2.6 节 3）和 4）定义。

6） 对任何选定的条目：

a） 对每个在 **rule-based-access-control** 下可能返回的属性类型，务必将访问赋予该类型的至少一个属性值；

b） 将不返回在 **rule-based-access-control** 下拒绝访问的属性值。

7） 对返回的信息应用 **basic-access-control**，如第 10.2.6 节 5）定义。

11 号码簿修改操作

有四种操作可用于修改号码簿：分别是在第 11.1 节- 第 11.4 节中定义的 **addEntry**、**removeEntry**、**modifyEntry** 和 **modifyDN**。

注 1 — 这些操作中的每一个都通过其不同的名称来确定目标条目。

注 2 — **addEntry**、**removeEntry** 和 **modifyDN** 操作的成功执行可能依赖于跨越号码簿的 DIB 的物理分布。如果失败，将报告带问题 **affectsMultipleDSAs** 的 **updateError**。见 ITU-T X.518 建议书| ISO/IEC 9594-4。

注 3 — 在基本通信机制发生故障的情况下，操作的结果是不确定的。用户应使用号码簿查询操作来检查尝试的修改操作是否取得了成功。

11.1 增加条目

11.1.1 增加条目语法

addEntry 操作用于向 DIT 增加一个叶条目（一个对象条目或一个别名条目）。操作变量可以由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

```
addEntry  OPERATION  ::= {
        ARGUMENT          AddEntryArgument
        RESULT            AddEntryResult
        ERRORS            { attributeError | nameError | serviceError | referral | securityError |
                          updateError }
        CODE              id-opcode-addEntry }

AddEntryArgument  ::=  OPTIONALLY-PROTECTED {
        SET {
                object                  [0]   Name,
                entry                   [1]   SET OF Attribute,
                targetSystem            [2]   AccessPoint  OPTIONAL,
                COMPONENTS OF                 CommonArguments} }

AddEntryResult  ::=  CHOICE {
        null              NULL,
        information       OPTIONALLY-PROTECTED-SEQ {
                          SEQUENCE { COMPONENTS OF CommonResultsSeq } } }
```

11.1.2 增加条目变量

object 变量用于确定将要增加的条目。其直接上级（为了操作取得成功，它必须已经存在）通过移去最后一个 RDN 分量（它属于将要创建的条目）来确定。**object** 可以是一个可选的名称，并可以包括正文信息，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 9.3 节所述。最后一个 RDN 分量将是主要的 RDN，并将包括所有的不同值，其所有属性的正文清单对 RDN 起作用。如果最后一个 RDN 分量中提供的任何 **AttributeTypeAndDistinguishedValue** 都不带可选的不同值，那么提供的单个值将用作该属性的单个不同值。

entry 变量包含属性信息，与来自 RDN 的信息一起组成将要创建的条目。号码簿将确保条目复合号码簿方案要求。如果正在创建的条目是一个别名，那么不需要任何检查即可确保 **aliasedEntryName** 属性指向一个有效的条目。

targetSystem 变量指明 DSA 持有新的条目。如果该变量不存在，那么它将意味着同一 DSA 持有新对象的上级。如果该变量出现，那么它将是带 **AccessPoint** 的 DSA。当增加子条目时，参数不存在。

如果变量存在，那么将对 **CommonArguments** 中 **criticalExtensions** 参数中的 **targetSystem** 位进行设置，指明该扩展是重要的。

注 1 — 如果指明或暗指 DSA 的选择与本地管理策略冲突，那么不执行操作并返回一个错误。

CommonArguments（见第 7.3 节）包括有关服务控制和适用于请求的安全参数的规定。除非在 **criticalExtensions** 中设置 **useAliasOnUpdate** 重要扩展位，否则忽略 **dontDereferenceAlias** 选项（并当作已经设置了）。因此，只有当不设置 **dontDereferenceAlias** 并且设置 **useAliasOnUpdate** 时，才通过该操作废弃别名。如果提供，那么忽略 **sizeLimit** 分量。如果请求方对该操作的变量进行标记、加密或者标记和加密，那么 **SecurityParameters**（见第 7.10 节）分量将包括在变量中。

注 2 — 如果遇到第一版本 DSA，那么涉及废弃别名的更新操作将总失败。

11.1.3 增加条目结果

如果请求成功，那么将返回一个结果。如果号码簿对该结果进行标记、加密或者标记和加密，那么 **CommonResultsSeq**（见第 7.4 节）的 **SecurityParameters** 分量（见第 7.10 节）将包括在结果中。如果号码簿不对该操作的结果进行标记，那么不随结果传送任何信息。

11.1.4 增加条目错误

如果请求失败，那么将报告其中一个列出的错误。对将报告特殊错误的情况在第 12 节中进行定义。

11.1.5 基本访问控制的增加条目操作决策点

如果 **rule-based-access-control** 也应用，那么有关 **basic-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **basic-access-control** 对正在增加的条目起作用，那么应用以下访问控制序列：

1） 对由 **object** 变量确定的条目的直接上级不需要任何特殊的许可。

注 1 — 安全策略可以防止号码簿用户越过 DSA 边界来增加条目（例如，利用 **targetSystem** 变量）。在这种情况下，可能返回一个适当的 **nameError**、**serviceError**、**securityError** 或 **updateError**，前提是它不会危害直接上级条目的存在。如果这样（即未将 *DiscloseOnError* 赋予上级条目），那么之后将执行第 7.11.3 节中定义的、关于上级条目的程序将紧跟其后。

2） 如果条目已经存在，并且不同名称等于 **object** 变量，那么依据第 11.1.5.1 节 a），该操作失败。

3） 对正在增加的新条目需要增加许可。如果不赋予该许可，那么依据第 11.1.5.1 节 b），该操作失败。

注 2 — 当尝试增加一个条目时，增加许可将作为 **prescriptiveACI** 提供，当尝试增加一个子条目时，增加许可将作为 **prescriptiveACI** 或 **subentryACI** 提供。

4） 对将要增加的每个属性类型和每个值，需要增加许可。如果未出现任何许可，那么依据第 11.1.5.1 节 c），该操作失败。

11.1.5.1 错误返回

如果操作失败，如第 11.1.5 节定义，那么应用以下程序：

a） 如果操作失败，如第 11.1.5 节 2）定义，那么有效的错误返回为以下之一：如果 *DiscloseOnError* 或增加许可赋予给了现有的条目，那么将返回一个带问题 **entryAlreadyExists** 的 **updateError**。否则对正在增加的条目，紧接着执行如第 7.11.3 节所述的程序。

b） 如果操作失败，如第 11.1.5 节 3）定义，那么对正在增加的条目，紧接着执行如第 7.11.3 节所述的程序。

c） 如果操作失败，如第 11.1.5 节 4）定义，那么有效的错误返回为带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**。

11.1.6 基于规则的访问控制的增加条目操作决策点

如果 **basic-access-control** 也应用，那么有关 **rule-based-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **rule-based-access-control**、**rule-and-basic-access-control** 或 **rule-and-simple-access-control** 对正在执行 **addEntry** 操作的那部分 DIB 起作用，那么应用以下访问控制序列：

1） 如果拒绝赋予直接上级基于规则的条目级许可，那么返回带问题 **noSuchObject** 的 **nameError**，如第 7.11.2.4 节定义。

2） 应用 **basic-access-control**，如第 11.1.5 节定义。

11.2 移去条目

11.2.1 移去条目语法

移去条目操作用于从 DIT 删去一个叶条目（一个对象条目、族成员或一个别名条目）或一个非叶祖先及其下属。操作变量可以由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

```
removeEntry  OPERATION  ::= {
        ARGUMENT            RemoveEntryArgument
        RESULT              RemoveEntryResult
        ERRORS              { nameError | serviceError | referral | securityError | updateError }
        CODE                id-opcode-removeEntry }

RemoveEntryArgument  ::=  OPTIONALLY-PROTECTED {
        SET {
                object                  [0]   Name,
                COMPONENTS OF                 CommonArguments } }

RemoveEntryResult  ::=  CHOICE {
        null                    NULL,
        information             OPTIONALLY-PROTECTED-SEQ {
                SEQUENCE  { COMPONENTS OF CommonResultsSeq } } }
```

11.2.2 移去条目变量

object 变量用于确定将要删去的条目。**object** 可以是一个可选的名称，并可以包括正文信息，如 ITU-T X.501 建议书| ISO/IEC 9594-2 第 9.3 节所述。

CommonArguments（见第 7.3 节）包括有关服务控制和适用于请求的安全参数的规定。除非在 **criticalExtensions** 中设置 **useAliasOnUpdate** 重要扩展位，否则忽略 **dontDereferenceAlias** 选项（并当作已经设置了）。因此，只有当不设置 **dontDereferenceAlias** 并且设置 **useAliasOnUpdate** 时，才通过该操作废弃别名。如果提供，那么忽略 **sizeLimit** 分量。如果请求方对该操作的变量进行标记、加密或者标记和加密，那么 **SecurityParameters**（见第 7.10 节）分量将包括在变量中。

注 — 如果遇到第一版本 DSA，那么涉及废弃别名的更新操作将总失败。

FamilyGrouping 可以按如下设置：

— 对该操作，**entryOnly** 为缺省值。将要移去的条目将是一个叶条目。

— 可以为祖先规定 **compoundEntry**。将移去复合条目的所有成员。如果目标对象不是一个祖先，那么操作失败，返回带问题 **notAncestor** 的 **updateError**。如果不可能移去所有成员，例如出于安全原因，那么操作也将失败，返回一个适当的错误。

如果 **FamilyGrouping** 不出现或设置为上述值之外的任何其他值，那么采用 **entryOnly**。

11.2.3 移去条目结果

如果请求成功，那么将返回一个结果。如果号码簿对该结果进行标记、加密或者标记和加密，那么 **CommonResultsSeq**（见第 7.4 节）的 **SecurityParameters** 分量（见第 7.10 节）将包括在结果中。如果号码簿不对该操作的结果进行标记，那么不随结果传送任何信息。

当 **EntryInformationSelection** 中的 **familyReturn** 选择族信息时，返回的信息在第 7.6.4 节中定义。

information 分量中返回的信息对应（成功）执行修改条目操作后的 DIB 状态。

11.2.4 移去条目错误

如果请求失败，那么将报告其中一个列出的错误。对将报告特殊错误的情况在第 12 节中进行定义。

11.2.5 基本访问控制的移去条目操作决策点

如果 **rule-based-access-control** 也应用，那么有关 **basic-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **basic-access-control** 对正在移去的条目起作用，那么应用以下访问控制：

— 对正在移去的条目需要移去许可。如果不赋予该许可，那么依据第 7.11.1 节，该操作失败。

注 — 对出现在被移去条目中的任何属性和属性值，不需要任何特殊的许可。

11.2.6 基于规则的访问控制的移去条目操作决策点

如果 **basic-access-control** 也应用，那么有关 **rule-based-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **rule-based-access-control**、**rule-and-basic-access-control** 或 **rule-and-simple-access-control** 对正在移去的条目起作用，那么应用以下访问控制序列：

1） 如果未赋予目标条目基于规则的条目级许可，那么操作失败，返回带问题 **noSuchObject** 的 **nameError**，如第 7.11.2.4 节定义。

2） 应用条目级 **basic-access-control**，如第 11.2.5 节规定。

3） 如果未赋予属性值基于规则的访问，那么它将不被移去。

4） 如果未赋予基于规则的 RDN 访问，那么 RDN 的任何属性值都将不被移去。如果移去所有的属性值，那么从条目中移去属性。如果移去所有的属性，那么从 DIT 中移去条目。如果移去至少一个属性值，并且请求方没有 RDN 许可，那么操作成功，但条目继续留在 DIT 中，带一个或多个属性。

注 1 — 除非条目不同值的标签正文的所有值都有相同的值，否则这不支持基于规则的访问控制策略。

5） 在 **rule-based-access-control** 下，如果赋予了 RDN 许可，但未赋予访问至少一个其他属性值的许可，那么不移去 RDN，操作失败，返回带问题 **insufficientAccessRights** 的 **securityError**。是否移去请求方拥有访问许可的其他属性值是一个局部问题。

注 2 — 这向请求方表明，至少存在一个无法访问的属性值。

6） 如果移去条目的所有属性，那么从 DIT 中移去该条目，操作成功。

11.3 修改条目

11.3.1 修改条目语法

修改条目操作用于对单个条目执行一系列以下修改中的一个或多个：

a） 增加一个新的属性；

b） 移去一个属性；

c） 增加属性值；

d） 移去属性值；

e） 替换属性值；

f） 修改一个别名；

g） 增加一个常量或一个属性的所有值；

h） 删去所有属性值，在每种情况下其撤退值为 **FALSE**。

操作变量可以由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

```
modifyEntry  OPERATION  ::= {
        ARGUMENT        ModifyEntryArgument
        RESULT          ModifyEntryResult
        ERRORS          { attributeError | nameError | serviceError | referral | securityError |
                        updateError }
        CODE            id-opcode-modifyEntry }

ModifyEntryArgument  ::=  OPTIONALLY-PROTECTED {
        SET {
                object                  [0]     Name,
                changes                 [1]     SEQUENCE OF EntryModification,
                selection               [2]     EntryInformationSelection OPTIONAL,
                COMPONENTS OF   CommonArguments } }

ModifyEntryResult  ::=  CHOICE {
        null                            NULL,
        information                     OPTIONALLY-PROTECTED-SEQ  {
                SEQUENCE {
                        entry                   [0]     EntryInformation OPTIONAL,
                        COMPONENTS OF           CommonResultsSeq } } }

EntryModification   ::=     CHOICE {
        addAttribute            [0]     Attribute,
        removeAttribute         [1]     AttributeType,
        addValues               [2]     Attribute,
        removeValues            [3]     Attribute,
        alterValues             [4]     AttributeTypeAndValue,
        resetValue              [5]     AttributeType,
        replaceValues           [6]     Attribute }
```

11.3.2 修改条目变量

object 变量用于确定适用于修改的条目。**object** 可以是一个可选的名称，并可以包括正文信息，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 9.3 节所述。

changes 变量用于确定修改序列，它以规定的次序应用。如果任何一个单个修改失败，那么产生一个 **attributeError**，条目仍处于操作之前它的状态。也就是说，操作是很小的。修改序列的最终结果不应与号码簿方案出现冲突。不过，对单个 **EntryModification** 修改它可能这样做，并且有时候需要这样做。可能发生以下类型的修改：

a） **addAttribute** — 它确定一个将要加入条目的新属性，它完全由变量确定。任何增加一个已经存在的属性的尝试都将导致一个 **attributeError**。

b） **removeAttribute** — 变量用于确定（通过其类型）一个将要从条目中移去的属性。任何移去一个不存在的属性的尝试都将导致一个 **attributeError**。

注 1 — 如果属性值出现在 RDN 中，那么将不允许进行本操作。

c） **addValues** — 通过变量中的变量类型，它确定一个属性，并规定一个或多个将要加入属性的属性值。任何增加一个已经存在的值的尝试都将导致一个错误。任何向一个不存在的类型增加一个值的尝试都将导致类型和值的增加。

d） **removeValues** — 通过变量中的变量类型，它确定一个属性，并规定一个或多个将要从属性中移去的属性值。如果值未出现在属性中，那么将导致一个 **attributeError**。任何从一个属性中移去最后一个值的尝试都将导致移去属性类型。

注 2 — 如果其中之一的值出现在 RDN 中，那么将不允许进行本操作。

可以通过或不通过一个正文清单来规定将要增加的属性或属性值。正文不能加入现有的属性值，不能从现有的属性值中移去，也不能修改。为了更改现有属性值的正文清单，首先需要移去属性值，而后以新的正文清单插入相同的属性值。当移去一个属性值时，将不提供任何正文清单，并且与正要移去的属性值相关的任何现有正文清单都将随属性值移去。

e） **alterValues** — 它确定一个属性类型，并规定一个将要加入所有属性值的数量。尝试对语法不是为 **INTEGER** 或 **REAL** 的属性进行修改将导致一个 **attributeError**。

f） **resetValue** — 它通过其类型确定一个属性，并移去属性的所有值（如果有的话），它有一个相关的属性值正文，其退路为 **FALSE**。**resetValue** 不移去任何没有正文的属性值。

g） **replaceValues** — 它用提供的值替换给定属性类型的所有现有值，如果它不存在，那么创建属性类型。如果它存在，不带值的替换将移去属性类型，如果类型不存在，将忽略之。

注 3 — 本号码簿规范不建立有关执行 DSA 对其所接收的 PDU 进行解码和处理的次序的规则。如果在处理每个元素之前，DSA 对整个 PDU 进行解码，并且如果对非可选的选项存在一个新的和非预期的值，如 **replaceValues**，那么 DSA 将发出一个编码错误信号。不过，如果 DSA 根据需要对各元素进行解码，那么它很可能将检测到一个未知的重要扩展，并返回一个不支持的重要扩展理由代码，告知操作失败。在任何一种情况下，DSA 不对操作进行处理都是正确的；不过，执行方应意识到，任何一种信号都可以用来指明操作失败。

变量将被单个 **ModifyEntry** 操作中的 **addValues** 和 **removeValues** 组合所替代。

CommonArguments（见第 7.3 节）包括有关服务控制和适用于请求的安全参数的规定。除非在 **criticalExtensions** 中设置 **useAliasOnUpdate** 重要扩展位，否则忽略 **dontDereferenceAlias** 选项（并当作已经设置了）。因此，只有当不设置 **dontDereferenceAlias** 并且设置 **useAliasOnUpdate** 时，才通过该操作废弃别名。如果提供，那么忽略 **sizeLimit** 分量。如果请求方对该操作的变量进行标记、加密或者标记和加密，那么 **SecurityParameters**（见第 7.10 节）分量将包括在变量中。

注 4 — 如果遇到第一版本 DSA，那么涉及废弃别名的更新操作将总失败。

selection 变量用于规定一个可选的条目信息选择，它用于控制是否在操作结果中返回信息，并规定返回的特定属性和值。只有当通过绑定操作商定的版本为 **v2** 或更高时，才规定它。

操作可用于修改号码簿操作属性。只能对那些不是分类 **noUserModification**（并且用户拥有有效的修改访问权限）的号码簿操作属性进行修改。

注 5 — 无论用户修改是否允许，号码簿都可以对号码簿操作属性的值进行修改，作为其他号码簿操作的副作用。

只有当服务控制 **subentries** 为 **TRUE**，并且 **object** 为实际持有待修改之联合属性的分条目时，操作才可用于修改联合属性。

注 6 — 因此，在修改读条目返回的信息时需小心：某些信息可能来自属性集，在针对条目本身的操作中不能对之进行修改。例如，不可能通过针对条目的 **removeAttribute** 条目修改来从一个（普通）条目中删去属性集（将返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**）。

如果值规定了辅助对象类别，那么操作可用于修改一个条目的对象类别属性值。不过，如果尝试修改一个对象类别值（它规定了一个条目的结构对象类别），那么将导致一个带问题 **objectClassModificationProhibited** 的 **updateError**。对辅助对象类别的任何修改将使超类链接与作为结果的对象类别定义保持一致与正确。

11.3.3 修改条目结果

如果请求成功，那么将返回一个 **result**。如果在操作变量中没有规定任何 **selection**，并且不对结果进行标记、加密或者标记和加密，那么返回空结果。如果没有规定任何 **selection**（但号码簿将对结果进行标记、加密或者标记和加密），那么省略条目分量。如果号码簿对结果进行标记、加密或者标记和加密，那么 **CommonResultsSeq**（见第 7.4 节）的 **SecurityParameters** 分量（见第 7.10 节）将包括在结果中。如果号码簿不对结果进行标记，那么不随结果传送任何条目信息。

11.3.4 修改条目错误

如果请求失败，那么将报告其中一个列出的错误。对将报告特殊错误的情况在第 12 节中进行定义。

11.3.5 基本访问控制的修改条目操作决策点

如果 **rule-based-access-control** 也应用，那么有关 **basic-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **basic-access-control** 对正在修改的条目起作用，那么应用以下访问控制序列：

1） 对正在修改的条目需要*修改*许可。如果不赋予该许可，那么依据第 7.11.1 节，该操作失败。

2） 对在序列中应用的、每个规定的 **EntryModification** 变量，需要以下许可：

i） 对在 **addAttribute** 参数中规定的属性类型和每个值的*增加*许可。如果不赋予这些许可或属性已经存在，那么依据第 11.3.5.1 节 a），该操作失败。

ii） 对在 **removeAttribute** 参数中规定的属性类型的*移去*许可。如果不赋予该许可，那么依据第 11.3.5.1 节 b），该操作失败。

注 1 — 对出现在被移去属性中的任何属性值，不需要任何特殊的许可。

iii） 对在 **addValues** 参数中规定的每个属性值的*增加*许可。如果不赋予这些许可或任何属性值已经存在，那么依据第 11.3.5.1 节 c），该操作失败。

iv） 对在 **removeValues** 参数中规定的每个值的*移去*许可。如果不赋予这些许可，那么依据第 11.3.5.1 节 d），该操作失败。

注 2 — 如果 removeValues 修改的最终结果是移去属性的最后一个值（它将移去属性自身），那么对规定的属性类型也需要移去许可。

v） 对在 **alterValues** 参数中规定的每个值的*增加*和*移去*许可。如果不赋予这些许可，那么依据第 11.3.5.1 节 e），该操作失败。

vi） 对将要通过 **resetValue** 参数移去的每个值的*移去*许可。如果将至少移去一个值并且不赋予这些许可，那么依据第 11.3.5.1 节 f），该操作失败。

11.3.5.1 错误返回

如果操作失败，如第 11.3.5 节定义，那么应用以下程序：

a） 如果操作失败，如第 11.3.5 节 第 2）条第 i）子条所定义，那么有效的错误返回是以下之一：如果属性已经存在，并且将 *discloseOnError* 或*增加*赋予了该属性，那么将返回一个带问题 **attributeOrValueAlreadyExists** 的 **attributeError**；否则，将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**。

b） 如果操作失败，如第 11.3.5 节 第 2）条第 ii）子条所定义，那么有效的错误返回是以下之一：如果将 *DiscloseOnError* 许可赋予了正要移去的属性，并且该属性存在，那么将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**；否则，将返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

c） 如果操作失败，如第 11.3.5 节第 2）条第 iii）子条所定义，那么有效的错误返回是以下之一：如果属性值已经存在，并且将 *discloseOnError* 或*增加*赋予了该属性值，那么将返回一个带问题 **attributeOrValueAlreadyExists** 的 **attributeError**；否则，将在属性级上对 *discloseOnError* 许可进行验证。如果将 *discloseOnError* 赋予了该属性，那么将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**；否则，将返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

d） 如果操作失败，如第 11.3.5 节 第 2）条第 iv）子条所定义，那么有效的错误返回是以下之一：如果将 *DiscloseOnError* 许可赋予了任何正要移去的属性值，那么将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**；否则，将返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

e） 如果操作失败，如第 11.3.5 节 第 2）条第 v）子条所定义，那么有效的错误返回是以下之一：如果将 *DiscloseOnError* 许可赋予了任何正要更改的属性值，那么将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**；否则，将返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

f） 如果操作失败，如第 11.3.5 节 第 2）条第 vi）子条所定义，那么有效的错误返回是以下之一：如果将 *DiscloseOnError* 许可赋予了任何正要移去的属性值，那么将返回一个带问题 **insufficientAccessRights** 或 **noInformation** 的 **securityError**；否则，将返回一个带问题 **noSuchAttributeOrValue** 的 **attributeError**。

11.3.6 基于规则的访问控制的修改条目操作决策点

如果 **basic-access-control** 也应用，那么有关 **rule-based-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **rule-based-access-control**、**rule-and-basic-access-control** 或 **rule-and-simple-access-control** 对正在修改的条目起作用，那么应用以下访问控制序列：

1） 如果未赋予目标条目基于规则的条目级许可，那么依据第 7.11.2.4 节，操作失败，返回带问题 **noSuchObject** 的 **nameError**。

2） 依据第 11.3.5.1 节，应用条目级 **basic-access-control**。

3） 务必准予对每个移去的属性值（如果有的话）进行访问。如果不将 **rule-based-access-control** 许可赋予任何将要移去的属性值，那么操作失败，返回带问题 **noSuchAttributeOrValue** 的 **attributeError**。

4） 依据第 11.3.5 节 2），应用属性级 **basic-access-control**。

11.4 修改DN

11.4.1 修改DN语法

修改 DN 操作用于修改条目的相对不同名称、修改条目的主要相对不同名称、增加和减少属性的不同值，与/或将条目移至 DIT 的一个新上级。它可以与对象条目一起使用，包括复合条目或别名条目。

对族成员，其使用限于以下情况，即受影响的族成员仍将留在同一复合条目中。

如果条目有下属，那么相应地对所有下属进行重新命名或移动（即子树仍保持完整）。请求方可以对操作变量进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 17.3 节）。如果这样请求，那么号码簿可以对结果进行标记、加密或者标记和加密。

注 1 — 第一版本系统只能将该操作用于修改叶条目的相对不同名称。

注 2 — 只有当旧的上级、新的上级、条目及其所有下属都在一个 DSA 中时，第二版本和后续版本系统才能使用该操作将条目移至一个新的上级。

注 3 — 操作不将条目移至一个新的上级；所有的条目都将继续留在最初的 DSA 中。

注 4 — 整体上操作成功或失败；某些条目移动、某些条目不移动将不算失败。对号码簿的用户，操作的任何中间状态都将是外部不可见的。

注 5 — 在该操作后可能需要某些离线行为，以便保持一致性，例如对任何持有不同名称值的条目的属性进行更新，指的是重新命名或移去的各条目。

注 6 — 对重新命名或移去条目的下属条目，不对其 **modifyTimeStamp** 属性进行更新。

```
modifyDN  OPERATION  ::= {
      ARGUMENT          ModifyDNArgument
      RESULT            ModifyDNResult
      ERRORS            { nameError | serviceError | referral | securityError | updateError }
      CODE              id-opcode-modifyDN }

ModifyDNArgument  ::=  OPTIONALLY-PROTECTED {
      SET {
            object              [0]     DistinguishedName,
            newRDN              [1]     RelativeDistinguishedName,
            deleteOldRDN        [2]     BOOLEAN DEFAULT FALSE,
            newSuperior         [3]     DistinguishedName OPTIONAL,
            COMPONENTS OF               CommonArguments } }

ModifyDNResult  ::=  CHOICE {
      null                    NULL,
      information             OPTIONALLY-PROTECTED-SEQ  {
            SEQUENCE {
                  newRDN                    RelativeDistinguishedName,
                  COMPONENTS OF             CommonResultsSeq } } }
```

11.4.2 修改DN 变量

object 变量用于确定将对其不同名称进行修改的条目。将不废弃名称中的各别名。**object** 可以是一个可选的名称，并可包括正文信息，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 9.3 节所述。

newRDN 变量用于规定条目新的 RDN。如果操作将条目移至一个新的上级，而不改变其 RDN，那么为该参数提供旧的 RDN。

如果在条目中尚未存在新的 RDN 属性值（作为旧的 RDN 的一部分或者作为非不同值），那么增加之。如果不能增加，那么返回一个错误。

对每个对 RDN 起作用的属性，如果不同的值通过正文区分，那么 **newRDN** 可以提供可选的不同值，如 ITU-T X.501 建议书| ISO/IEC 9594-2 第 9.3 节所述。如果这样，那么 **newRDN** 将是一个主要 RDN，并将包括所有的不同值，其针对所有属性的正文清单对 RDN 起作用（包括继续保留作为不同值的现有的不同值）。不带可选不同值而提供的、**newRDN** 中的一个 **AttributeTypeAndDistinguishedValue** 用于指明有关该属性的一个单个不同值。

如果设置了 **deleteOldRDN** 标志，那么不在新的 RDN 中的旧的 RDN 中的所有属性值都将被删去。这包括通过正文区分的可选不同值，如果它们存在于旧的 RDN 中而未包括在新的 RDN 中。如果未设置该标志，那么旧的不同值仍将留在条目中（但不再是不同值）。当通过操作改变了 RDN 中的一个单个值属性，那么将设置该标志。如果旧的 RDN 中的属性值等同于新的 RDN 中的属性值（除了其正文清单），那么旧的 RDN 中的属性值将被新的 RDN 中的属性值所替换。如果该操作移去了属性的最后一个属性值，那么该属性将被删去。

如果出现，那么 **newSuperior** 规定条目将被移至 DIT 中的一个新的上级。条目变成为带指明不同名称的条目的一个直接下属，它必须是一个已经存在的对象条目。新的上级不会是条目自身，或者任何其下属，或者一个别名，否则移去的条目将与任何 DIT 结构规则都将产生冲突。移去条目的条目下属有可能与活动的子方案产生冲突，在这种情况下，子方案管理权威部门负责对这些条目进行后续调整，使之与子方案保持一致，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 14 节所述。

如果变量存在，那么将对 **CommonArguments** 中 **criticalExtensions** 参数中的 **newSuperior** 位进行设置，指明该扩展是重要的。

newSuperior 可以是一个可选的名称，并可以包括正文信息，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 第 9.3 节所述。

CommonArguments（见第 7.3 节）包括有关服务控制和适用于请求的安全参数的规定。出于该操作的目的，**dontDereferenceAlias** 选项与 **sizeLimit** 分量不相关，如果提供，将被忽略。该操作从不废弃别名。如果请求方对该操作的变量进行标记、加密或者标记和加密，那么 **SecurityParameters**（见第 7.10 节）分量将包括在变量中。

11.4.3 修改DN 结果

如果请求成功，那么将返回一个结果。如果号码簿对该结果进行标记、加密或者标记和加密，那么 **CommonResultsSeq**（见第 7.4 节）的 **SecurityParameters** 分量（见第 7.10 节）以及新的 RDN 将包括在结果中。如果号码簿不对结果进行标记，那么不随结果传送任何信息。

11.4.4 修改DN 错误

如果请求失败，那么将报告其中一个列出的错误。对将报告特殊错误的情况在第 12 节中进行定义。

11.4.5 基本访问控制的修改DN 操作

如果 **rule-based-access-control** 也应用，那么有关 **basic-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **basic-access-control** 对正在重新命名的条目起作用，那么应用以下访问控制：

— 如果操作的作用是修改条目的 RDN，那么对正在重新命名的条目需要*重新命名*许可（考虑其最初的名称）。如果不赋予该许可，那么依据第 11.4.5.1 节，操作失败。

— 如果操作的作用是将一个条目移至 DIT 的一个新上级，那么对正在考虑其最初名称的条目需要*输出*许可，对正在考虑其新名称的条目需要*输入*许可。如果不赋予这些许可中的任何一个许可，那么依据第 11.4.5.1 节，操作失败。

注 1 — 输入许可应作为说明性 ACI 提供。

注 2 — 无需任何额外的许可，即使作为修改名称最后 RDN 的结果，需要增加一个新的不同值或移去一个旧的值。

11.4.5.1 错误返回

如果操作失败，如第 11.4.5 节所定义，那么紧跟着将执行第 7.11.1 节中所述的程序，它有关重新命名的条目（考虑其最初的名称）。

11.4.6 基于规则的访问控制的修改DN 操作

如果 **basic-access-control** 也应用，那么有关 **rule-based-access-control** 应用的次序问题将是一个局部问题，除非如果任何一个机制都拒绝对条目、属性类型或属性值的访问，那么它将不被其他机制覆盖。在这种情况下，**basic-access-control** 的 *DiscloseOnError* 许可是一种将不覆盖 **rule-based-access-control** 拒绝的许可。

如果 **rule-based-access-control**、**rule-and-basic-access-control** 或 **rule-and-simple-access-control** 对正在重新命名的条目起作用，那么应用以下访问控制序列：

1） 如果未赋予目标条目基于规则的 RDN 许可，那么依据第 7.11.2.4 节，操作失败，返回带问题 **noSuchObject** 的 **nameError**。

2） 依据第 11.4.5 节，应用条目级 **basic-access-control**。

3） 如果操作的作用是将条目移至 DIT 的一个新上级，那么对新的上级需要基于规则的 RDN 许可，否则依据第 7.11.2.4 节，操作失败，返回带问题 **noSuchObject** 的 **nameError**。

12 错误

12.1 错误优先权

号码簿不继续执行以下点之外的操作，即它在该点确定报告错误。

注 1 — 该规则的一个含义是，对相同查询的重复例子，遇到的第一个错误可以不同，原因是，对处理某个给定的查询，没有一个特定的逻辑次序。例如，可以以不同的次序来搜索 DSA。

注 2 — 此处规定的错误优先权规则仅适用于号码簿作为一个整体提供的抽象服务。当考虑号码簿的内部结构时，将应用不同的规则。

如果号码簿同时检测到多个错误，那么以下清单将确定报告哪个错误。清单中级别较高的错误比级别较低的错误具有更高的逻辑优先权，报告级别较高的错误。

a） **nameError**；

b） **updateError**；

c） **attributeError**；

d） **securityError**；

e） **serviceError**。

以下错误不会引起任何优先权冲突：

a） **abandonFailed**，原因是它只针对放弃这一操作，不会遇到任何其他错误；

b） **abandoned**，如果与错误检测同时接收到放弃操作，那么不报告它。在这种情况下，返回带问题 **tooLate** 的 **abandonFailed**，并报告遇到的实际错误；

c） **referral**，它不是一个“真实的”错误，只是指出号码簿已经检测到 DUA 应将其请求提交给另一个访问点。

12.2 放弃

如果 DUA 利用适当的 **InvokeId** 调用一个放弃操作，那么对任何未完成的号码簿查询操作（即读、搜索、比较、列表）都可以报告该结果。如果操作参数由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节），那么号码簿可以对错误参数进行标记、加密或者标记和加密。

```
abandoned ERROR ::= {   -- 不是字面上的“错误”
        PARAMETER          OPTIONALLY-PROTECTED {
                SET {COMPONENTS OF     CommonResults} }
        CODE          id-errcode-abandoned }
```

如果号码簿对错误进行标记、加密或者标记和加密，那么 **SecurityParameters** 分量（见第 7.10 节）将包括在 **CommonResults** 中（见第 7.4 节）。

12.3 放弃失败

abandonFailed 错误用于报告在尝试放弃一个操作过程中遇到的问题。如果操作参数由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节），那么号码簿可以对错误参数进行标记、加密或者标记和加密。

```
abandonFailed ERROR ::= {
        PARAMETER                  OPTIONALLY-PROTECTED {
                SET {
                        problem                 [0]          AbandonProblem,
                        operation               [1]          InvokeId,
                        COMPONENTS OF           CommonResults } }
        CODE                       id-errcode-abandonFailed }

AbandonProblem ::= INTEGER { noSuchOperation (1), tooLate (2), cannotAbandon (3) }
```

各个参数具有以下含义。

规定了遇到的特殊 **problem**。可能指出任何以下问题：

a） **noSuchOperation** — 当号码簿不了解将被放弃的操作时（可能由于未发生此类调用，或由于号码簿忘了它）；

b） **tooLate** — 当号码簿已对操作做出响应时；

c） **cannotAbandon** — 当已尝试放弃一个操作时（对之是禁止的，如修改），或者当无法执行放弃时。

将放弃确认特定的 **operation**（调用）。

如果号码簿对错误进行标记、加密或者标记和加密，**SecurityParameters** 分量（见第 7.10 节）将包括在 **CommonResults** 中（见第 7.4 节）

通过使用 **CommonResults** 的 **notification** 分量，可以任选地对错误问题提供的信息做出限制。

12.4 属性错误

attributeError 用于报告一个与属性有关的问题。如果操作参数由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节），那么号码簿可以对错误参数进行标记、加密或者标记和加密。

```
attributeError ERROR ::= {
        PARAMETER          OPTIONALLY-PROTECTED {
                SET {
                        object          [0]     Name,
                        problems        [1]     SET OF SEQUENCE {
                                problem         [0]     AttributeProblem,
                                type            [1]     AttributeType,
                                value           [2]     AttributeValue  OPTIONAL },
                        COMPONENTS OF CommonResults } }
        CODE            id-errcode-attributeError }
```

```
AttributeProblem  ::=  INTEGER {
        noSuchAttributeOrValue              (1),
        invalidAttributeSyntax              (2),
        undefinedAttributeType              (3),
        inappropriateMatching               (4),
        constraintViolation                 (5),
        attributeOrValueAlreadyExists       (6),
        contextViolation                    (7) }
```

各个参数具有以下含义。

object 参数用于确定当发生错误时操作正在使用哪个条目。返回的名称可以只包括有关包含多个不同值（由正文区分）的属性的主要不同值（即 DSA 不必使用正文选择，如第 7.7 节所述，如同它对成功的操作所做的那样）。

可以规定一个或多个 **problems**。每个 **problem**（在下面确定）伴随一个属性 **type** 指示，如果需要避免模糊性，那么 **value** 将引起以下问题：

a） **noSuchAttributeOrValue** — 命名的条目缺少一个属性或属性值，它规定为操作的一个变量。

b） **invalidAttributeSyntax** — 规定为操作的一个变量的假设属性值不符合属性类型的属性语法。

c） **undefinedAttributeType** — 提供了一个未定义的属性类型，作为操作的一个变量。该错误只可能发生在与 **addEntry** 或 **modifyEntry** 相关的操作中。

d） **inappropriateMatching** — 尝试使用一个未为相关属性类型定义的匹配规则，如在一个过滤器中。

e） **constraintViolation** — 在操作变量中提供的属性值不符合 ITU-T X.501 建议书 | ISO/IEC 9594-2 或属性定义所提的约束要求（例如，值超过了允许的最大值）。

f） **attributeOrValueAlreadyExists** — 尝试增加一个已经在条目中存在的属性，或者一个已经在属性中存在的值。

g） **contextViolation** — 在操作变量中随属性值提供的正文清单或正文不符合 ITU-T X.501 建议书 | ISO/IEC 9594-2 或正文定义（例如，正文值不符合正确的语法）或 DIT 正文使用所提的约束要求。

如果号码簿对错误进行标记、加密或者标记和加密，**SecurityParameters** 分量（见第 7.10 节）将包括在 **CommonResults** 中（见第 7.4 节）

通过使用 **CommonResults** 的 **notification** 分量，可以任选地对错误问题提供的信息做出限制。

12.5 名称错误

nameError 用于报告一个与名称有关的问题，提供的名称作为操作的变量。如果操作参数由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书| ISO/IEC 9594-2 的第 17.3 节），那么号码簿可以对错误参数进行标记、加密或者标记和加密。

```
nameError  ERROR  ::=  {
      PARAMETER       OPTIONALLY-PROTECTED {
            SET {
                  problem                 [0]     NameProblem,
                  matched                 [1]     Name,
                  COMPONENTS OF           CommonResults } }
      CODE                id-errcode-nameError }

NameProblem  ::=  INTEGER {
      noSuchObject                        (1),
      aliasProblem                        (2),
      invalidAttributeSyntax              (3),
      aliasDereferencingProblem           (4),
      contextProblem                      (5) }
```

各个参数具有以下含义。

遇到特殊的 **problem**。可能指出任何以下问题：

a） **noSuchObject** — 提供的名称不匹配任何对象的名称。

b） **aliasProblem** — 别名已废弃，它未命名任何对象。

c） **invalidAttributeSyntax** — 名称中 AVA 中的属性类型及其伴随的属性值不兼容。

d） **aliasDereferencingProblem** — 在不允许别名的地方或者拒绝访问的地方遇到了别名。

e） **contextProblem** — 名称中所用的正文类型或值无法理解或无效、正文变量名称的使用不可接受，或者在名称解析期间，一个假设的名称匹配于多个 DIT 条目的名称。

matched 参数包含匹配的 DIT 中最低条目（对象或别名）的名称，并且是所提供名称的简短形式，或者如果已经废弃别名，那么是结果名称的简短形式。返回的名称可以只包括有关包含多个不同值（由正文区分）的属性的主要不同值（即 DSA 不必使用正文选择，如第 7.7 节所述，如同它对成功的操作所做的那样）。

注 — 如果在号码簿操作变量提供的名称中存在属性类型与/或值问题，那么它通过带问题 **invalidAttributeSyntax** 的 **nameError** 来报告，而不是作为 **attributeError** 或 **updateError** 。

如果号码簿对错误进行标记、加密或者标记和加密，**SecurityParameters** 分量（见第 7.10 节）将包括在 **CommonResults** 中（见第 7.4 节）

通过使用 **CommonResults** 的 **notification** 分量，可以任选地对错误问题提供的信息做出限制。

12.6 提名

referral 操作用于将服务用户重新引导至一个或多个更适于完成所请求操作的访问点上。如果操作参数由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书| ISO/IEC 9594-2 的第 17.3 节），那么号码簿可以对错误参数进行标记、加密或者标记和加密。

```
referral ERROR ::= {    -- 不是字面上的“错误”
    PARAMETER OPTIONALLY-PROTECTED {
        SET {
            candidate           [0]     ContinuationReference,
            COMPONENTS OF               CommonResults } }
    CODE                id-errcode-referral }
```

错误有一个单个参数，它包含一个 **ContinuationReference**，用于推动操作（见 ITU-T X.518 建议书| ISO/IEC 9594-4）。

如果 DSA 对应一个 LDAP 请求，那么 **ContinuationReference** 中的 **accessPoints** 分量将包含一个有效的 **LabeledURI** 值，在这种情况下，它将使用该值来创建一个 LDAP 提名。如果它不包含一个有效的 **LabeledURI** 值，那么它将不返回一个提名。

如果号码簿对错误进行标记，那么 **SecurityParameters** 分量（见第 7.10 节）将包括在 **CommonResults** 中（见第 7.4 节）。

在进行连续引用之前，DUA 将检查未将某个等同于将由连续引用产生的请求的请求作为处理相同用户请求的一部分。如果已经这样做了，那么 DUA 将不进行连续引用，以避免循环引用。

12.7 安全错误

securityError 用于报告在因安全原因而执行的操作中出现的问题。如果操作参数由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 17.3 节），那么号码簿可以对错误参数进行标记、加密或者标记和加密。

```
securityError ERROR ::= {
    PARAMETER       OPTIONALLY-PROTECTED {
        SET {
            problem             [0]  SecurityProblem,
            spkmInfo            [1]  SPKM-ERROR,
            COMPONENTS OF       CommonResults } }
```

```
        CODE                id-errcode-securityError }

SecurityProblem  ::=  INTEGER {
        inappropriateAuthentication     (1),
        invalidCredentials              (2),
        insufficientAccessRights        (3),
        invalidSignature                (4),
        protectionRequired              (5),
        noInformation                   (6),
        blockedCredentials              (7),
        invalidQOPMatch                 (8),
        spkmError                       (9) }
```

错误有一个单个参数，用于报告遇到的特殊 **problem**。可以指出以下问题：

a） **inappropriateAuthentication** — 与请求方证书相关的安全级别与请求的保护级别不一致，例如，提供的是简单的证书，而要求的是增强的证书。

b） **invalidCredentials** — 提供的证书无效。

c） **insufficientAccessRights** — 请求方无权完成请求的操作。

d） **invalidSignature** — 发现请求签名无效。

e） **protectionRequired** — 号码簿不愿完成请求的操作，原因是变量未标记。

f） **noInformation** — 请求的操作产生一个安全错误，对它没有任何信息可用。

g） **blockedCredentials** — 证书因安全方面的因素而受阻（例如，因连续多次提供无效的口令）。返回该错误的决定由对 DSA 起作用的安全策略进行管理。

h） **invalidQOPMatch** — 两个实体具有不同的保护参数，分别为各自的安全服务而定义。

i） **spkmError** — 发现提供的 SPKM 令牌无效。**spkmInfo** 参数包含一个指示，指出这是一个 SPKM 错误令牌，以及与该错误相关的 SPKM 正文标识符。

如果号码簿对错误进行标记、加密或者标记和加密，**SecurityParameters** 分量（见第 7.10 节）将包括在 **CommonResults** 中（见第 7.4 节）

通过使用 **CommonResults** 的 **notification** 分量，可以任选地对错误问题提供的信息做出限制。

12.8 服务错误

serviceError 用于报告与服务提供有关的问题。如果操作参数由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书| ISO/IEC 9594-2 的第 17.3 节），那么号码簿可以对错误参数进行标记、加密或者标记和加密。

```
serviceError  ERROR  ::=  {
        PARAMETER           OPTIONALLY-PROTECTED {
                SET {
                        problem                 [0]     ServiceProblem,
                        COMPONENTS OF                   CommonResults } }
        CODE                id-errcode-serviceError }

ServiceProblem  ::=  INTEGER {
        busy                            (1),
        unavailable                     (2),
        unwillingToPerform              (3),
        chainingRequired                (4),
        unableToProceed                 (5),
        invalidReference                (6),
        timeLimitExceeded               (7),
        administrativeLimitExceeded     (8),
        loopDetected                    (9),
```

```
unavailableCriticalExtension      (10),
outOfScope                        (11),
ditError                          (12),
invalidQueryReference             (13),
requestedServiceNotAvailable      (14),
unsupportedMatchingUse            (15),
ambiguousKeyAttributes            (16),
saslBindInProgress                (17) }
```

错误有一个单个参数，用于报告遇到的特殊 **problem**。可以指出以下问题：

a） **busy** — 号码簿或其某部分目前太忙，无法执行请求的操作，但可以在短暂等待后执行。

b） **unavailable** — 号码簿或其某部分目前无法使用。

c） **unwillingToPerform** — 号码簿或其某部分目前尚未做好执行该请求的准备，例如，由于它将导致过量耗费资源或者与所涉及的管理权威部门的策略存在冲突。

d） **chainingRequired** — 除了通过链接，号码簿无法完成请求，不过，链接被 **chainingProhibited** 服务控制选项所禁止。

e） **unableToProceed** — 返回该错误的 DSA 对相应的命名正文没有管理权限，结果是，无法参与名称解析。

f） **invalidReference** — 无法执行 DUA 指向的请求，（通过 **OperationProgress**） — 这可能因使用一个无效的提名而引起。

g） **timeLimitExceeded** — 号码簿到达了用户在服务控制中设置的时间限度。没有任何部分结果可返回给用户。

h） **administrativeLimitExceeded** — 号码簿到达了管理权威部门设置的某个限度，并且没有任何部分结果可返回给用户。

i） **loopDetected** — 因内部循环，号码簿无法完成该请求。

j） **unavailableCriticalExtension** — 因一个或多个重要扩展无法使用，号码簿无法完成请求。

k） **outOfScope** — 在请求范围内没有任何提名可用。

l） **ditError** — 因 DIT 一致性问题，号码簿无法完成请求。

m） **invalidQueryReference** — 请求操作的参数无效。如果分页结果中的 **queryReference** 无效，那么报告该问题。

注 — 第一版本系统不支持该问题。

n） **requestedServiceNotAvailable** — 由于没有任何搜索规则可用于搜索，或者由于搜索与所用的搜索规则发生冲突，因此服务特定管理区域内的搜索请求失败。可以与该服务问题一起返回额外的诊断信息。对不同情况下的此类额外信息在第 13 节中定义。

o） **unsupportedMatchingUse** — 当设置了 **performExactly** 搜索选项时，尝试使用 DSA 不支持的匹配规则，例如，在过滤器中。

p） **ambiguousKeyAttributes** — 选择了基于映射的匹配规则，但可映射的过滤器项提供了多个违犯相关映射表的匹配。该错误情况伴随一个通告属性，由相关的、基于匹配的匹配规则指明。

q） **saslBindInProgress** — 对某些认证机制，请求方可能需要多次调用 **directoryBind** 操作。这通过响应方发送一个带问题 **saslBindInProgress** 的 **serviceError** 来指明。它指出，响应方要求请求方调用一个新的 **directoryBind** 操作，利用同样的 **SaslCredentials** 机制，继续进行认证过程。如果请求方希望能在任何阶段中断认证过程，那么它可以调用一个 **SaslAbort** 设为 **TRUE** 的 **directoryBind** 操作。

如果号码簿对错误进行标记、加密或者标记和加密，**SecurityParameters** 分量（见第 7.10 节）将包括在 **CommonResults** 中（见第 7.4 节）

通过使用 **CommonResults** 的 **notification** 分量，可以任选地对错误问题提供的信息做出限制。

12.9 更新错误

updateError 用于报告与尝试增加、删除或修改 DIB 中信息有关的问题。如果操作参数由请求方进行标记、加密或者标记和加密（见 ITU-T X.501 建议书| ISO/IEC 9594-2 的第 17.3 节），那么号码簿可以对错误参数进行标记、加密或者标记和加密。

```
updateError ERROR ::= {
        PARAMETER       OPTIONALLY-PROTECTED {
                SET {
                        problem             [0]     UpdateProblem,
                        attributeInfo       [1]     SET SIZE (1..MAX) OF CHOICE {
                                attributeType               AttributeType,
                                attribute                   Attribute } OPTIONAL,
                        COMPONENTS OF  CommonResults } }
        CODE                        id-errcode-updateError }

UpdateProblem ::= INTEGER {
        namingViolation                         (1),
        objectClassViolation                    (2),
        notAllowedOnNonLeaf                     (3),
        notAllowedOnRDN                         (4),
        entryAlreadyExists                      (5),
        affectsMultipleDSAs                     (6),
        objectClassModificationProhibited       (7),
        noSuchSuperior                          (8),
        notAncestor                             (9),
        parentNotAncestor                       (10),
        hierarchyRuleViolation                  (11),
        familyRuleViolation                     (12) }
```

问题参数用于报告遇到的特殊问题。可以指出以下问题：

a） **namingViolation** — 所做的增加或修改尝试将与号码簿方案和 ITU X.510 建议书| ISO/IEC 9594-2 中定义的 DIT 结构规则发生冲突。也就是说，它将把一个条目作为别名条目的下属，或者在 DIT 区域内不允许其对象类别成员，或者将为条目定义一个 RDN，以便包括禁止的属性类型。

b） **objectClassViolation** — 所做的更新尝试将产生一个与条目内容规则不一致的条目，例如其对象类别定义、DTI 内容规则，或者与对象类别相关的、ITU-T X.501 建议书 | ISO/IEC 9594-2 中的各定义。

c） **notAllowedOnNonLeaf** — 只允许对 DIT 的叶条目尝试操作。

d） **notAllowedOnRDN** — 所做的操作尝试将对 RDN 产生影响（例如，移去作为 RDN 一部分的某个属性）。

e） **entryAlreadyExists** — 尝试的 **addEntry** 或 **modifyDN** 操作命名一个已经存在的条目。

注 1 — 这包括由 RDN 引起的冲突，它包括多个由正文区分的不同值，而不管正文是什么，如 ITU-T X.501 建议书| ISO/IEC 9594-2 所述。

f） **affectsMultipleDSAs** — 所做的更新尝试将需要在多个不允许该操作的 DSA 上进行。

g） **objectClassModificationProhibited** — 一个尝试修改条目结构对象类别的操作。

h） **noSuchSuperior** — 尝试的 **modifyDN** 操作命名一个不存在的、新的上级条目。

i） **notAncestor** — 一个尝试删除复合条目而不将祖先规定为对象的操作。

j） **parentNotAncestor** — 一个尝试将条目建成为非祖先族成员下直接层次下属的操作。

k） **hierarchyRuleViolation** — 一个尝试打破适用于层次型组的规则的操作：一个层次型组必须完全处于任何服务特定管理区域外，或者必须完全包含在某个服务特定管理区域内；层次型组局限于一个单个 DSA。

l） **familyRuleViolation** — 一个尝试打破适用于复合条目内各族的规则的操作。

attributeInfo 参数用于确定引起问题的特殊属性类型和可能值。如果报告 **objectClassViolation**，那么 **attribute** 项将出现，指明引起问题的 **objectClass** 属性类型，列出引起问题的对象类别；还可以出现额外的 **attributeType** 项（例如，用于确定丢失的强制性属性或外来属性）。

注 2 — **updateError** 不是利用 **addEntry**、**removeEntry**、**modifyEntry** 或 **modifyDN** 操作中遇到的属性类型、值或约束冲突来报告问题。此类问题通过 **attributeError** 来报告。

如果号码簿对错误进行标记、加密或者标记和加密，那么 **SecurityParameters** 分量（见第 7.10 节）将包括在 **CommonResults** 中（见第 7.4 节）。

通过使用 **CommonResults** 的 **notification** 分量，可以任选地对错误问题提供的信息做出限制。

13 分析搜索变量

本节只与在服务特定管理区域内开始其最初评估阶段的搜索操作有关。

本程序有两个目的：

a） 它提供了搜索确认功能（见 ITU-T X.501 建议书| ISO/IEC 9594-2 第 16.12 节）。不过，搜索确认功能不产生错误信息。如果在程序运行期间遇到错误，那么评估停止并返回 FALSE，否则返回 TRUE。对空搜索规则的搜索确认将总返回 TRUE。

b） 当未找到任何管理搜索规则时，使用本程序，当可能确定单个搜索规则时，可以对 **SearchArgument** 进行评估，以确定 **search** 请求为什么失败。在这种情况下，当发现一个错误条件时，评估停止，在 **CommonResults** 数据类型的 **notification** 分量中提供必要的诊断信息，并返回一个带问题 **requestedServiceNotAvailable** 的服务错误。包括什么样的诊断信息取决于确定的错误类型。

注 — 依据上述规定，可以利用相同的搜索规则对一个搜索请求做两次评估。如何优化它不在本规范的讨论范围内，而是一个执行决定。

本程序假设执行将不允许可调用的搜索规则：

— 规定不支持的属性类型、正文类型、匹配规则、匹配限制等；
— 规定基于映射的匹配算法，它们不支持或不相关搜索规则正管理的搜索类型；
— 规定与搜索规则冲突的匹配规则替换；
— 指的是执行不支持的、可选的搜索规则特性；或者
— 不一致的或错误的。

13.1 一般性检查搜索过滤器

评估首先检查过滤器是否与某些利用以下程序的基本限制有冲突：

1） 如果过滤器中描述的属性类型未由请求属性概貌在 **inputAttributeTypes** 搜索规则分量中进行描述，那么 **notification** 将包含：
   — 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchAttributeViolation**；
   — 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及
   — 一个 **attributeTypeList** 通告属性，值为用于确定非法属性类型的对象标识符。

2） 如果存在只通过求反的过滤器项描述的属性类型，那么 **notification** 将包含：
   — 一个 **searchServiceProblem** 通告属性，值为 **id-pr-attributeNegationViolation**；
   — 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及
   — 一个 **attributeTypeList** 通告属性，值为用于确定在过滤器中被非法求反的属性类型的对象标识符。

3） 检查 **attributeCombination** 中规定的条件是否满足有关属性类型非求反出现的要求。如果强制性属性类型，即需无条件地由非求反的过滤器项在过滤器中描述的属性类型，未出现在任何分过滤器中，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-missingSearchAttribute**；

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及

— 一个 **attributeTypeList** 通告属性，值为用于确定丢失属性类型的对象标识符。

如果要求的组合未出现，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchAttributeCombinationViolation**；

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及

— 一个用于确定丢失组合的 **attributeCombinations** 通告属性。

4） 对拥有 **selectedValues** 子分量而值集为空的请求属性概貌，检查这些属性类型中是否存在不满足以下要求之一的过滤器项：

— 过滤器项为 **present** 类型，并且 **contexts** 子分量未出现在请求属性概貌中；或者

— 过滤器项为 **contextPresent** 类型，并且 **contexts** 子分量出现在请求属性概貌中。

如果上述检查对任何过滤器项都失败，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchValueNotAllowed**；

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及

— 一个 **filterItem** 通告属性，值为失败的过滤器项。

5） 对拥有 **contexts** 子分量的请求属性概貌，检查正文类型中是否存在不包括在该子分量中的过滤器项。如果存在，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchContextViolation**；

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及

— 一个 **contextTypeList** 通告属性，值为有关非法正文类型的对象标识符。

6） 如果在搜索规则中采用了针对 **subset** 分量的 **allowed** 选项，那么检查 **SearchArgument** 的 **subset** 变量是否符合规定的要求。如果不复合，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchSubsetViolation**；以及

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量。

13.2 检查请求属性概貌

如果上述程序未发生任何错误，那么需对每个分过滤器进行检查，看各分过滤器中所描述的任何属性类型是否也都有效地出现了。该程序不规定任何对分过滤器进行评估的次序。对有效出现在分过滤器中的属性类型，至少需要通过一个非求反的过滤器项对其进行描述，指出它符合相应请求属性概貌的要求。利用以下程序对非求反的过滤器项进行评估。

按以下次序对非求反的过滤器项进行检查：

1） 对每个分过滤器，对需无条件进行描述的属性类型的过滤器项进行检查；

2） 对每个分过滤器，对需条件进行描述的属性类型的过滤器项进行检查；以及

3） 对每个分过滤器，对剩余的过滤器项进行检查。

如果对分过滤器的评估失败，那么评估停止并返回错误信息，详述如下。

如果分过滤器中的属性类型通过若干非求反的过滤器项进行描述，那么原则上对每个这样的过滤器项都进行检查，直至找到一个符合要求的过滤器项，或对所有的过滤器项都进行了检查。如果在程序运行过程中有一个过滤器项失败，那么它留待进一步评估。由最后一个失败的属性类型过滤器项来决定返回的诊断信息。

利用以下程度对过滤器项进行评估：

1） 如果请求属性概貌中的 **selectedValues** 分量不存在，或者它存在并非空，那么检查 **equality**、**substrings**、**approximateMatch** 或 **extensibleMatch** 类型的过滤器项。如果不这样，那么 **notification** 将包含：
   — 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchValueRequired**；
   — 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及
   — 一个 **attributeTypeList** 通告属性，值为用于从过滤器项中确定属性类型的对象标识符。
2） 如果相应请求属性概貌中的 **selectedValues** 子分量存在并非空，那么检查过滤器项是否不匹配该子分量中规定的任何值。如果这样，那么 **notification** 将包含：
   — 一个 **searchServiceProblem** 通告属性，值为 **id-pr-invalidSearchValue**；
   — 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及
   — 一个 **filterItem** 通告属性，失败的过滤器项作为其唯一值。
3） 如果 **contexts** 子分量未出现，那么继续下一个小节。
4） 检查 **contextCombination** 子分量中规定的条件是否满足有关正文类型的要求。如果强制性正文类型丢失，即需无条件地为属性类型描述的正文类型，那么 **notification** 将包含：
   — 一个 **searchServiceProblem** 通告属性，值为 **id-pr-missingSearchContext**；
   — 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；
   — 一个 **attributeTypeList** 通告属性，只有一个值，为用于从过滤器项中确定属性类型的对象标识符；以及
   — 一个 **contextTypeList** 通告属性，值为用于确定丢失正文类型的对象标识符。

   如果要求的组合未出现，那么 **notification** 将包含：
   — 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchContextCombinationViolation**；
   — 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；
   — 一个 **attributeTypeList** 通告属性，只有一个值，为用于从过滤器项中确定属性类型的对象标识符；以及
   — 一个 **contextCombinations** 通告属性，用于确定丢失的组合。
5） 检查分过滤器中的属性类型的正文命题是否全部都包括在了 **contexts** 子分量中。如果不这样，那么 **notification** 将包含：
   — 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchContextViolation**；
   — 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；
   — 一个 **attributeTypeList** 通告属性，只有一个值，为用于从过滤器项中确定属性类型的对象标识符；以及
   — 一个 **contextTypeList** 通告属性，值为用于确定属性类型不允许的正文类型的对象标识符。
6） 如果包括了请求属性概貌 **contexts** 子分量中任何正文类型的正文值，那么检查在分过滤器中为属性类型规定的任何正文命题是否都包含了未为 **contexts** 子分量中相应正文类型规定的值。如果这样，那么 **notification** 将包含：
   — 一个 **searchServiceProblem** 通告属性，值为 **id-pr-searchContextValueViolation**；
   — 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；

— 一个 **attributeTypeList** 通告属性，只有一个值，为用于从过滤器项中确定属性类型的对象标识符；以及

— 一个 **contextList** 通告属性，值为属性类型不允许的正文命题。

13.3 检查控制和层次选择

如果搜索请求未成功完成控制和层次选择的测试，如 ITU-T X.501 建议书 | ISO/IEC 9594-2 中所规定，那么将执行本小节中的程序。

1） 如果搜索规则的 **defaultControls** 分量或 **defaultControls** 的 **hierarchyOptions** 子分量不存在，并且搜索请求规定 **self** 旁的层次选择，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-hierarchySelectForbidden**；以及

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量。

2） 如果依据搜索规则中 **defaultControls** 和 **mandatoryControls** 分量的组合，不允许请求中的层次选择选项或丢失某些选择，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-invalidHierarchySelect**；

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及

— 一个 **hierarchySelectList** 通告属性，值为确定无效层次选择选项的比特串。

3） 如果 DSA 不支持请求中的层次选择选项，并且不被 2）涵盖，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-unavailableHierarchySelect**；

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及

— 一个 **hierarchySelectList** 通告属性，值为确定不支持层次选择选项的比特串。

4） 如果依据搜索规则中 **defaultControls** 和 **mandatoryControls** 分量的组合，不允许请求中的搜索控制选项（通过第 10.2.1 节定义）或丢失某些选项，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-invalidSearchControlOptions**；

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及

— 一个 **searchControlOptionsList** 通告属性，值为确定无效搜索控制选项的比特串。

5） 如果依据搜索规则中 **defaultControls** 和 **mandatoryControls** 分量的组合，不允许请求中的服务控制选项或丢失某些选项，那么 **notification** 将包含：

— 一个 **searchServiceProblem** 通告属性，值为 **id-pr-invalidServiceControlOptions**；

— 一个 **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；以及

— 一个 **serviceControlOptionsList** 通告属性，值为确定无效服务控制选项的比特串。

13.4 检查匹配使用

在搜索确认程序中，该本小节代表确认中的最后一步，它假设 **search** 请求已通过了所有其他确认步骤。最后一步失败的搜索规则置于 ITU-T X.518 建议书 | ISO/IEC 9594-4 的 **MatchProblemSR** 清单中（见第 19.3.2.2.1 节的第 3 项）。

如果搜索请求不符合 **matchingUse** 的要求，如 ITU-T X.501 建议书| ISO/IEC 9594-2 第 16.10.2 节中有关请求属性概貌的规定，那么有关其中之一失败的请求属性概貌的 **notification** 将包含：

— 如果匹配限制冲突，那么是一个带值 **id-pr-attributeMatchingViolation** 的 **searchServiceProblem** 通告属性，如果匹配规则以一种不支持的方式进行应用，那么是一个带值 **id-pr-unsupportedMatchingUse** 的 **searchServiceProblem** 通告属性；

— **serviceType** 通告属性，值为搜索规则的 **serviceType** 分量；

— **attributeTypeList** 通告属性，值只为确定属性类型的对象标识符；以及

— 对冲突的匹配限制，应包含额外的通告属性，由规范为该匹配限制规定。

注 — 当有若干 request-attribute-profiles 无法确认时，那么选择哪个来创建一个 **notification** 将是一件本地的事情。

附 件 A

ASN.1 中的抽象服务

（本附件是本建议书 | 国际标准的组成部分）

本附件包括本号码簿规范中所含的所有 ASN.1 类型、值和信息对象定义，形式为 ASN.1 模块 **DirectoryAbstractService**。

```
DirectoryAbstractService {joint-iso-itu-t ds(5) module(1) directoryAbstractService(2) 5}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --
-- 输出本模块中定义的类型和值用于号码簿规范中所含的其他 ASN.1 模块，并供其他将使用它们
-- 访问号码簿服务的应用使用。其他应用可以将它们用于其自身目的，但这不会限制维护或改进号码簿
-- 服务所需的扩展和修改。

IMPORTS

-- 来自 ITU-T X.501 建议书| ISO/IEC 9594-2

    attributeCertificateDefinitions, authenticationFramework, basicAccessControl,
    commonProtocolSpecification, directoryShadowAbstractService, distributedOperations,
    enhancedSecurity, id-at, informationFramework, selectedAttributeTypes, serviceAdministration,
    upperBounds
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}

    Attribute, ATTRIBUTE, AttributeType, AttributeTypeAssertion, AttributeValue,
    AttributeValueAssertion, CONTEXT, ContextAssertion, DistinguishedName,
    MATCHING-RULE, Name, OBJECT-CLASS, RelativeDistinguishedName,
    SupportedAttributes, SupportedContexts
        FROM  InformationFramework informationFramework

    RelaxationPolicy
        FROM  ServiceAdministration serviceAdministration

    AttributeTypeAndValue
        FROM BasicAccessControl  basicAccessControl

    OPTIONALLY-PROTECTED{ }, OPTIONALLY-PROTECTED-SEQ{ }
        FROM EnhancedSecurity enhancedSecurity

-- 来自 ITU-T X.518 建议书| ISO/IEC 9594-4

    AccessPoint, ContinuationReference, Exclusions, OperationProgress, ReferenceType
        FROM  DistributedOperations  distributedOperations

-- 来自 ITU-T X.519 建议书| ISO/IEC 9594-5

    Code, ERROR, id-errcode-abandoned, id-errcode-abandonFailed, id-errcode-attributeError,
    id-errcode-nameError, id-errcode-referral, id-errcode-securityError, id-errcode-serviceError,
    id-errcode-updateError, id-opcode-abandon, id-opcode-addEntry, id-opcode-compare,
    id-opcode-list,  id-opcode-modifyDN, id-opcode-modifyEntry, id-opcode-read,
    id-opcode-removeEntry, id-opcode-search, InvokeId, OPERATION
        FROM CommonProtocolSpecification commonProtocolSpecification

-- 来自 ITU-T X.520 建议书| ISO/IEC 9594-6

    DirectoryString {}
        FROM SelectedAttributeTypes selectedAttributeTypes
```

```
        ub-domainLocalID, ub-saslMechanism
                FROM UpperBounds upperBounds

-- 来自 ITU-T X.509 建议书| ISO/IEC 9594-8

        AlgorithmIdentifier, CertificationPath, ENCRYPTED {}, SIGNATURE {}, SIGNED {}
                FROM AuthenticationFramework authenticationFramework


        AttributeCertificationPath
                FROM AttributeCertificateDefinitions attributeCertificateDefinitions

-- 来自 ITU-T X.525 建议书| ISO/IEC 9594-9

        AgreementID
                FROM DirectoryShadowAbstractService  directoryShadowAbstractService

-- 来自 RFC 2025

        SPKM-ERROR, SPKM-REP-TI, SPKM-REQ
                FROM SpkmGssTokens { iso (1) identified-organization (3) dod(6) internet (1)
                        security (5) mechanisms (5) spkm (1) spkmGssTokens (10) }      ;

-- 公共数据类型 --

CommonArguments  ::=  SET {
        serviceControls          [30]   ServiceControls DEFAULT { },
        securityParameters       [29]   SecurityParameters OPTIONAL,
        requestor                [28]   DistinguishedName OPTIONAL,
        operationProgress        [27]   OperationProgress
                                            DEFAULT { nameResolutionPhase notStarted },
        aliasedRDNs              [26]   INTEGER OPTIONAL,
        criticalExtensions       [25]   BIT STRING  OPTIONAL,
        referenceType            [24]   ReferenceType  OPTIONAL,
        entryOnly                [23]   BOOLEAN DEFAULT TRUE,
        nameResolveOnMaster      [21]   BOOLEAN DEFAULT FALSE,
        operationContexts        [20]   ContextSelection OPTIONAL,
        familyGrouping           [19]   FamilyGrouping DEFAULT entryOnly }

FamilyGrouping  ::=  ENUMERATED  {
        entryOnly           (1),
        compoundEntry       (2),
        strands             (3),
        multiStrand         (4) }

CommonResults  ::=  SET {
        securityParameters  [30]   SecurityParameters                    OPTIONAL,
        performer           [29]   DistinguishedName                     OPTIONAL,
        aliasDereferenced   [28]   BOOLEAN                               DEFAULT FALSE,
        notification        [27]   SEQUENCE SIZE (1..MAX) OF Attribute   OPTIONAL }

CommonResultsSeq  ::=  SEQUENCE {
        securityParameters  [30]   SecurityParameters                    OPTIONAL,
        performer           [29]   DistinguishedName                     OPTIONAL,
        aliasDereferenced   [28]   BOOLEAN                               DEFAULT FALSE,
        notification        [27]   SEQUENCE SIZE (1..MAX) OF Attribute   OPTIONAL }

ServiceControls  ::=  SET {
        options                         [0]    ServiceControlOptions DEFAULT { },
        priority                        [1]    INTEGER { low (0),  medium (1),  high (2) }  DEFAULT medium,
        timeLimit                       [2]    INTEGER  OPTIONAL,
        sizeLimit                       [3]    INTEGER  OPTIONAL,
        scopeOfReferral                 [4]    INTEGER { dmd(0), country(1) } OPTIONAL,
        attributeSizeLimit              [5]    INTEGER  OPTIONAL,
        manageDSAITPlaneRef             [6]    SEQUENCE {
                dsaName                                  Name,
                agreementID                              AgreementID } OPTIONAL,
        serviceType                     [7]    OBJECT IDENTIFIER OPTIONAL,
        userClass                       [8]    INTEGER OPTIONAL }
```

```
ServiceControlOptions  ::=  BIT STRING {
                preferChaining              (0),
                chainingProhibited          (1),
                localScope                  (2),
                dontUseCopy                 (3),
                dontDereferenceAliases      (4),
                subentries                  (5),
                copyShallDo                 (6),
                partialNameResolution       (7),
                manageDSAIT                 (8),
                noSubtypeMatch              (9),
                noSubtypeSelection          (10),
                countFamily                 (11),
                dontSelectFriends           (12),
                dontMatchFriends            (13) }

EntryInformationSelection  ::=  SET  {
        attributes                          CHOICE {
                allUserAttributes                   [0]     NULL,
                select                              [1]     SET OF AttributeType
                -- 空集意味着不需要任何属性 -- } DEFAULT allUserAttributes : NULL,
        infoTypes               [2]         INTEGER {
                attributeTypesOnly                          (0),
                attributeTypesAndValues                     (1) } DEFAULT attributeTypesAndValues,
        extraAttributes                     CHOICE {
                allOperationalAttributes            [3]     NULL,
                select                              [4]     SET SIZE (1..MAX) OF AttributeType } OPTIONAL,
        contextSelection                    ContextSelection OPTIONAL,
        returnContexts                      BOOLEAN DEFAULT FALSE,
        familyReturn                        FamilyReturn  DEFAULT
                                            { memberSelect  contributingEntriesOnly } }

ContextSelection  ::=  CHOICE  {
        allContexts             NULL,
        selectedContexts        SET SIZE (1..MAX) OF TypeAndContextAssertion }

TypeAndContextAssertion  ::=  SEQUENCE  {
        type                    AttributeType,
        contextAssertions       CHOICE {
                preference              SEQUENCE OF ContextAssertion,
                all                     SET OF ContextAssertion } }

FamilyReturn ::= SEQUENCE {
        memberSelect        ENUMERATED {
                contributingEntriesOnly     (1),
                participatingEntriesOnly    (2),
                compoundEntry               (3) },
        familySelect        SEQUENCE SIZE (1..MAX) OF OBJECT-CLASS.&id   OPTIONAL }

EntryInformation  ::=  SEQUENCE  {
        name                                        Name,
        fromEntry                                   BOOLEAN DEFAULT TRUE,
        information                                 SET SIZE (1..MAX) OF CHOICE {
                attributeType                               AttributeType,
                attribute                                   Attribute } OPTIONAL,
        incompleteEntry             [3]     BOOLEAN DEFAULT FALSE, -- 不在第一版本系统中
        partialName                 [4]     BOOLEAN DEFAULT FALSE, -- 不在第一或第二版本系统中
        derivedEntry                [5]     BOOLEAN DEFAULT FALSE -- 不在第四版本之前的系统中 -- }

family-information  ATTRIBUTE  ::=  {
        WITH SYNTAX             FamilyEntries
        USAGE                   directoryOperation
        ID                      id-at-family-information }

FamilyEntries  ::=  SEQUENCE  {
        family-class            OBJECT-CLASS.&id,       -- 结构对象类别值
        familyEntries           SEQUENCE OF FamilyEntry }
```

```
FamilyEntry ::= SEQUENCE {
        rdn                     RelativeDistinguishedName,
        information             SEQUENCE OF CHOICE {
                attributeType           AttributeType,
                attribute               Attribute },
        family-info             SEQUENCE SIZE (1..MAX) OF FamilyEntries OPTIONAL }

Filter ::= CHOICE {
        item    [0]     FilterItem,
        and     [1]     SET OF Filter,
        or      [2]     SET OF Filter,
        not     [3]     Filter }

FilterItem      ::=     CHOICE {
        equality                [0]     AttributeValueAssertion,
        substrings              [1]     SEQUENCE {
                type                            ATTRIBUTE.&id ({ SupportedAttributes }),
                strings                         SEQUENCE OF CHOICE {
                        initial                 [0]     ATTRIBUTE.&Type
                                                        ({SupportedAttributes}{@substrings.type}),
                        any                     [1]     ATTRIBUTE.&Type
                                                        ({SupportedAttributes}{@substrings.type}),
                        final                   [2]     ATTRIBUTE.&Type
                                                        ({SupportedAttributes}{@substrings.type}),
                        control                         Attribute } },      -- 用于规定以下项的解释
        greaterOrEqual  [2]     AttributeValueAssertion,
        lessOrEqual     [3]     AttributeValueAssertion,
        present         [4]     AttributeType,
        approximateMatch [5]    AttributeValueAssertion,
        extensibleMatch [6]     MatchingRuleAssertion,
        contextPresent  [7]     AttributeTypeAssertion }

MatchingRuleAssertion ::= SEQUENCE {
        matchingRule    [1]     SET SIZE (1..MAX) OF MATCHING-RULE.&id,
        type            [2]     AttributeType  OPTIONAL,
        matchValue      [3]     MATCHING-RULE.&AssertionType ( CONSTRAINED BY {
                        -- matchValue 应是一个类型值，由 matchingRule 确定的其中一个
                        -- MATCHING-RULE 的&AssertionType 字段规定。-- } ),
        dnAttributes    [4]     BOOLEAN DEFAULT FALSE }

PagedResultsRequest ::= CHOICE {
        newRequest              SEQUENCE {
                pageSize                INTEGER,
                sortKeys                SEQUENCE SIZE (1..MAX) OF SortKey OPTIONAL,
                reverse         [1]     BOOLEAN DEFAULT FALSE,
                unmerged        [2]     BOOLEAN DEFAULT FALSE,
                pageNumber      [3]     INTEGER OPTIONAL },
        queryReference          OCTET STRING,
        abandonQuery    [0]     OCTET STRING }

SortKey ::= SEQUENCE {
        type            AttributeType,
        orderingRule    MATCHING-RULE.&id OPTIONAL }

SecurityParameters ::= SET {
        certification-path      [0]     CertificationPath               OPTIONAL,
        name                    [1]     DistinguishedName               OPTIONAL,
        time                    [2]     Time                            OPTIONAL,
        random                  [3]     BIT STRING                      OPTIONAL,
        target                  [4]     ProtectionRequest               OPTIONAL,
        response                [5]     BIT STRING                      OPTIONAL,
        operationCode           [6]     Code                            OPTIONAL,
        attributeCertificationPath [7]  AttributeCertificationPath      OPTIONAL,
        errorProtection         [8]     ErrorProtectionRequest          OPTIONAL,
        errorCode               [9]     Code                            OPTIONAL }

ProtectionRequest ::= INTEGER { none (0), signed (1) }
```

```
Time ::= CHOICE {
        utcTime                 UTCTime,
        generalizedTime         GeneralizedTime }

ErrorProtectionRequest ::= INTEGER { none (0), signed (1) }

-- 绑定和解开操作 --

directoryBind OPERATION ::= {
        ARGUMENT        DirectoryBindArgument
        RESULT          DirectoryBindResult
        ERRORS          { directoryBindError } }

DirectoryBindArgument ::= SET {
        credentials             [0]     Credentials OPTIONAL,
        versions                [1]     Versions DEFAULT {v1} }

Credentials ::= CHOICE {
        simple                  [0]     SimpleCredentials,
        strong                  [1]     StrongCredentials,
        externalProcedure       [2]     EXTERNAL,
        spkm                    [3]     SpkmCredentials,
        sasl                    [4]     SaslCredentials }

SimpleCredentials ::= SEQUENCE {
        name                    [0]     DistinguishedName,
        validity                [1]     SET {
                time1                   [0]     CHOICE {
                        utc                     UTCTime,
                        gt                      GeneralizedTime } OPTIONAL,
                time2                   [1]     CHOICE {
                        utc                     UTCTime,
                        gt                      GeneralizedTime } OPTIONAL,
                random1                 [2]     BIT STRING  OPTIONAL,
                random2                 [3]     BIT STRING  OPTIONAL } OPTIONAL,
        password                [2]     CHOICE {
                unprotected             OCTET STRING,
                protected               SIGNATURE {OCTET STRING} } OPTIONAL}

StrongCredentials ::= SET {
        certification-path              [0]     CertificationPath OPTIONAL,
        bind-token                      [1]     Token,
        name                            [2]     DistinguishedName OPTIONAL,
        attributeCertificationPath      [3]     AttributeCertificationPath OPTIONAL }

SpkmCredentials ::= CHOICE {
        req     [0]     SPKM-REQ,
        rep     [1]     SPKM-REP-TI }

SaslCredentials ::=  SEQUENCE {
        mechanism               [0]     DirectoryString { ub-saslMechanism },
        credentials             [1]     OCTET STRING OPTIONAL,
        saslAbort               [2]     BOOLEAN DEFAULT FALSE }

Token ::= SIGNED { SEQUENCE {
        algorithm               [0]     AlgorithmIdentifier,
        name                    [1]     DistinguishedName,
        time                    [2]     UTCTime,
        random                  [3]     BIT STRING,
        response                [4]     BIT STRING OPTIONAL,
        bindIntAlgorithm        [5]     SEQUENCE SIZE (1..MAX) OF AlgorithmIdentifier OPTIONAL,
        bindIntKeyInfo          [6]     BindKeyInfo OPTIONAL,
        bindConfAlgorithm       [7]     SEQUENCE SIZE (1..MAX) OF AlgorithmIdentifier OPTIONAL,
        bindConfKeyInfo         [8]     BindKeyInfo OPTIONAL } }

Versions ::= BIT STRING {v1(0), v2(1) }

DirectoryBindResult ::= DirectoryBindArgument
```

```
directoryBindError ERROR ::= {
        PARAMETER                OPTIONALLY-PROTECTED {
                SET {
                        versions                [0]     Versions DEFAULT {v1},
                        error                           CHOICE {
                                serviceError                    [1]     ServiceProblem,
                                securityError                   [2]     SecurityProblem } } } }

BindKeyInfo ::= ENCRYPTED { BIT STRING }
```

-- 操作、变量和结果 --

```
read OPERATION ::= {
        ARGUMENT        ReadArgument
        RESULT          ReadResult
        ERRORS          { attributeError | nameError | serviceError | referral | abandoned |
                        securityError }
        CODE            id-opcode-read }

ReadArgument ::= OPTIONALLY-PROTECTED {
        SET {
                object                  [0]     Name,
                selection               [1]     EntryInformationSelection DEFAULT { },
                modifyRightsRequest     [2]     BOOLEAN DEFAULT FALSE,
                COMPONENTS OF                   CommonArguments } }

ReadResult ::= OPTIONALLY-PROTECTED {
        SET {
                entry           [0]     EntryInformation,
                modifyRights    [1]     ModifyRights OPTIONAL,
                COMPONENTS OF   CommonResults } }

ModifyRights ::= SET OF SEQUENCE {
        item                            CHOICE {
                        entry                   [0]     NULL,
                        attribute               [1]     AttributeType,
                        value                   [2]     AttributeValueAssertion },
        permission [3]      BIT STRING { add (0), remove (1), rename (2), move (3) } }

compare OPERATION ::= {
        ARGUMENT        CompareArgument
        RESULT          CompareResult
        ERRORS          { attributeError | nameError | serviceError | referral | abandoned |
                        securityError }
        CODE            id-opcode-compare }

CompareArgument ::= OPTIONALLY-PROTECTED {
        SET {
                object          [0]     Name,
                purported       [1]     AttributeValueAssertion,
                COMPONENTS OF           CommonArguments } }

CompareResult ::= OPTIONALLY-PROTECTED {
        SET {
                name                            Name OPTIONAL,
                matched                 [0]     BOOLEAN,
                fromEntry               [1]     BOOLEAN DEFAULT TRUE,
                matchedSubtype          [2]     AttributeType OPTIONAL,
                COMPONENTS OF                   CommonResults } }

abandon OPERATION ::= {
        ARGUMENT AbandonArgument
        RESULT   AbandonResult
        ERRORS   { abandonFailed }
        CODE     id-opcode-abandon }

AbandonArgument ::= OPTIONALLY-PROTECTED-SEQ {
        SEQUENCE {
                invokeID        [0]     InvokeId } }
```

```
AbandonResult ::= CHOICE {
        null            NULL,
        information     OPTIONALLY-PROTECTED-SEQ {
                                SEQUENCE {
                                        invokeID                InvokeId,
                                        COMPONENTS OF           CommonResultsSeq } } }

list    OPERATION ::= {
        ARGUMENT        ListArgument
        RESULT          ListResult
        ERRORS          { nameError | serviceError | referral | abandoned | securityError }
        CODE            id-opcode-list }

ListArgument ::= OPTIONALLY-PROTECTED {
        SET {
                object                  [0]     Name,
                pagedResults            [1]     PagedResultsRequest OPTIONAL,
                listFamily              [2]     BOOLEAN DEFAULT FALSE,
                COMPONENTS OF           CommonArguments } }

ListResult ::= OPTIONALLY-PROTECTED {
        CHOICE {
                listInfo                        SET {
                        name                                    Name  OPTIONAL,
                        subordinates            [1]     SET OF SEQUENCE {
                                rdn                                     RelativeDistinguishedName,
                                aliasEntry              [0]     BOOLEAN DEFAULT  FALSE,
                                fromEntry               [1]     BOOLEAN DEFAULT  TRUE },
                        partialOutcomeQualifier [2]     PartialOutcomeQualifier OPTIONAL,
                        COMPONENTS OF                   CommonResults },
                uncorrelatedListInfo    [0]     SET OF ListResult } }

PartialOutcomeQualifier ::= SET {
        limitProblem                    [0]     LimitProblem  OPTIONAL,
        unexplored                      [1]     SET SIZE (1..MAX) OF ContinuationReference  OPTIONAL,
        unavailableCriticalExtensions   [2]     BOOLEAN DEFAULT  FALSE,
        unknownErrors                   [3]     SET SIZE (1..MAX) OF ABSTRACT-SYNTAX.&Type  OPTIONAL,
        queryReference                  [4]     OCTET STRING  OPTIONAL,
        overspecFilter                  [5]     Filter OPTIONAL,
        notification                    [6]     SEQUENCE SIZE (1 .. MAX) OF Attribute  OPTIONAL,
        entryCount                              CHOICE {
                bestEstimate                    [7] INTEGER,
                lowEstimate                     [8] INTEGER,
                exact                           [9]     INTEGER } OPTIONAL,
        streamedResult                  [10]    BOOLEAN DEFAULT  FALSE }

LimitProblem ::= INTEGER {
        timeLimitExceeded (0), sizeLimitExceeded (1), administrativeLimitExceeded (2) }

search  OPERATION ::= {
        ARGUMENT        SearchArgument
        RESULT          SearchResult
        ERRORS          { attributeError | nameError | serviceError | referral | abandoned |
                        securityError }
        CODE            id-opcode-search }

SearchArgument ::= OPTIONALLY-PROTECTED {
        SET {
                baseObject              [0]     Name,
                subset                  [1]     INTEGER {
                        baseObject(0), oneLevel(1), wholeSubtree(2) } DEFAULT baseObject,
                filter                  [2]     Filter DEFAULT  and : { },
                searchAliases           [3]     BOOLEAN  DEFAULT TRUE,
                selection               [4]     EntryInformationSelection DEFAULT { },
                pagedResults            [5]     PagedResultsRequest OPTIONAL,
                matchedValuesOnly       [6]     BOOLEAN DEFAULT FALSE,
                extendedFilter          [7]     Filter OPTIONAL,
                checkOverspecified      [8]     BOOLEAN DEFAULT FALSE,
                relaxation              [9]     RelaxationPolicy    OPTIONAL,
                extendedArea            [10]    INTEGER             OPTIONAL,
```

```
            hierarchySelections     [11]    HierarchySelections DEFAULT { self },
            searchControlOptions    [12]    SearchControlOptions    DEFAULT { searchAliases },
            joinArguments           [13]    SEQUENCE SIZE (1..MAX) OF JoinArgument OPTIONAL,
            joinType                [14]    ENUMERATED {
                    innerJoin(0), leftOuterJoin(1), fullOuterJoin(2) } DEFAULT leftOuterJoin,
            COMPONENTS OF           CommonArguments } }

HierarchySelections  ::=  BIT STRING {
        self                (0),
        children            (1),
        parent              (2),
        hierarchy           (3),
        top                 (4),
        subtree             (5),
        siblings            (6),
        siblingChildren     (7),
        siblingSubtree      (8),
        all                 (9) }

SearchControlOptions  ::=  BIT STRING {
        searchAliases                   (0),
        matchedValuesOnly               (1),
        checkOverspecified              (2),
        performExactly                  (3),
        includeAllAreas                 (4),
        noSystemRelaxation              (5),
        dnAttribute                     (6),
        matchOnResidualName             (7),
        entryCount                      (8),
        useSubset                       (9),
        separateFamilyMembers           (10),
        searchFamily                    (11) }

JoinArgument  ::=  SEQUENCE {
        joinBaseObject          [0]     Name,
        domainLocalID           [1]     DomainLocalID OPTIONAL,
        joinSubset              [2]     ENUMERATED {
                        baseObject(0), oneLevel(1), wholeSubtree(2) } DEFAULT baseObject,
        joinFilter              [3]     Filter  OPTIONAL,
        joinAttributes          [4]     SEQUENCE SIZE (1..MAX) OF JoinAttPair OPTIONAL,
        joinSelection           [5]     EntryInformationSelection }

DomainLocalID       ::= DirectoryString { ub-domainLocalID }

JoinAttPair  ::=  SEQUENCE {
        baseAtt                 AttributeType,
        joinAtt                 AttributeType,
        joinContext             SEQUENCE SIZE (1..MAX) OF JoinContextType OPTIONAL }

JoinContextType  ::=  CONTEXT.&id({SupportedContexts})

SearchResult  ::=  OPTIONALLY-PROTECTED {
        CHOICE {
            searchInfo                  SET {
                name                                Name  OPTIONAL,
                entries                     [0]     SET OF EntryInformation,
                partialOutcomeQualifier     [2]     PartialOutcomeQualifier OPTIONAL,
                altMatching                 [3]     BOOLEAN  DEFAULT FALSE,
                COMPONENTS OF                       CommonResults },
            uncorrelatedSearchInfo  [0]     SET OF SearchResult } }

addEntry  OPERATION  ::=  {
        ARGUMENT        AddEntryArgument
        RESULT          AddEntryResult
        ERRORS          { attributeError | nameError | serviceError | referral | securityError |
                        updateError }
        CODE            id-opcode-addEntry }
```

```
AddEntryArgument ::= OPTIONALLY-PROTECTED {
        SET {
                object                  [0]     Name,
                entry                   [1]     SET OF Attribute,
                targetSystem            [2]     AccessPoint  OPTIONAL,
                COMPONENTS OF   CommonArguments} }

AddEntryResult ::= CHOICE {
        null                    NULL,
        information             OPTIONALLY-PROTECTED-SEQ {
                                SEQUENCE { COMPONENTS OF CommonResultsSeq } } }

removeEntry  OPERATION  ::= {
        ARGUMENT        RemoveEntryArgument
        RESULT          RemoveEntryResult
        ERRORS          { nameError | serviceError | referral | securityError | updateError }
        CODE            id-opcode-removeEntry }

RemoveEntryArgument ::= OPTIONALLY-PROTECTED {
        SET {
                object                  [0]     Name,
                COMPONENTS OF           CommonArguments } }

RemoveEntryResult ::= CHOICE {
        null            NULL,
        information  OPTIONALLY-PROTECTED-SEQ {
                SEQUENCE { COMPONENTS OF CommonResultsSeq } } }

modifyEntry  OPERATION  ::= {
        ARGUMENT        ModifyEntryArgument
        RESULT          ModifyEntryResult
        ERRORS          { attributeError | nameError | serviceError | referral | securityError |
                        updateError }
        CODE            id-opcode-modifyEntry }

ModifyEntryArgument ::= OPTIONALLY-PROTECTED {
        SET {
                object                  [0]     Name,
                changes                 [1]     SEQUENCE OF EntryModification,
                selection               [2]     EntryInformationSelection OPTIONAL,
                COMPONENTS OF           CommonArguments } }

ModifyEntryResult ::= CHOICE {
        null                    NULL,
        information             OPTIONALLY-PROTECTED-SEQ  {
                SEQUENCE {
                        entry           [0]     EntryInformation OPTIONAL,
                        COMPONENTS OF   CommonResultsSeq } } }

EntryModification  ::=  CHOICE {
        addAttribute        [0]     Attribute,
        removeAttribute     [1]     AttributeType,
        addValues           [2]     Attribute,
        removeValues        [3]     Attribute,
        alterValues         [4]     AttributeTypeAndValue,
        resetValue          [5]     AttributeType,
        replaceValues       [6]     Attribute }

modifyDN  OPERATION  ::= {
        ARGUMENT        ModifyDNArgument
        RESULT          ModifyDNResult
        ERRORS          { nameError | serviceError | referral | securityError | updateError }
        CODE            id-opcode-modifyDN }

ModifyDNArgument ::= OPTIONALLY-PROTECTED {
        SET {
                object                  [0]     DistinguishedName,
                newRDN                  [1]     RelativeDistinguishedName,
                deleteOldRDN            [2]     BOOLEAN DEFAULT FALSE,
                newSuperior             [3]     DistinguishedName OPTIONAL,
```

```
            COMPONENTS OF           CommonArguments } }

ModifyDNResult ::= CHOICE {
        null                NULL,
        information   OPTIONALLY-PROTECTED-SEQ  {
            SEQUENCE {
                newRDN                  RelativeDistinguishedName,
                COMPONENTS OF           CommonResultsSeq } } }

-- 错误和参数 --

abandoned ERROR ::= {          -- 不是字面上的“错误”
        PARAMETER       OPTIONALLY-PROTECTED {
            SET {COMPONENTS OF      CommonResults} }
        CODE            id-errcode-abandoned }

abandonFailed ERROR ::= {
        PARAMETER               OPTIONALLY-PROTECTED {
            SET {
                problem         [0]     AbandonProblem,
                operation       [1]     InvokeId,
                COMPONENTS OF           CommonResults } }
        CODE            id-errcode-abandonFailed }

AbandonProblem ::= INTEGER { noSuchOperation (1), tooLate (2), cannotAbandon (3) }

attributeError ERROR ::= {
        PARAMETER       OPTIONALLY-PROTECTED {
            SET {
                object      [0]     Name,
                problems    [1]     SET OF SEQUENCE {
                    problem     [0]     AttributeProblem,
                    type        [1]     AttributeType,
                    value       [2]     AttributeValue  OPTIONAL },
                COMPONENTS OF CommonResults } }
        CODE            id-errcode-attributeError }

AttributeProblem ::= INTEGER {
        noSuchAttributeOrValue          (1),
        invalidAttributeSyntax          (2),
        undefinedAttributeType          (3),
        inappropriateMatching           (4),
        constraintViolation             (5),
        attributeOrValueAlreadyExists   (6),
        contextViolation                (7) }

nameError ERROR ::= {
        PARAMETER       OPTIONALLY-PROTECTED {
            SET {
                problem         [0]     NameProblem,
                matched         [1]     Name,
                COMPONENTS OF           CommonResults } }
        CODE            id-errcode-nameError }

NameProblem ::= INTEGER {
        noSuchObject                    (1),
        aliasProblem                    (2),
        invalidAttributeSyntax          (3),
        aliasDereferencingProblem       (4),
        contextProblem                  (5) }

referral ERROR ::= {    -- 不是字面上的“错误”
        PARAMETER OPTIONALLY-PROTECTED {
            SET {
                candidate       [0]     ContinuationReference,
                COMPONENTS OF           CommonResults } }
        CODE            id-errcode-referral }
```

```
securityError  ERROR  ::=  {
        PARAMETER          OPTIONALLY-PROTECTED {
                SET {
                        problem                 [0]     SecurityProblem,
                        spkmInfo                [1]     SPKM-ERROR,
                        COMPONENTS OF           CommonResults } }
        CODE                    id-errcode-securityError }

SecurityProblem  ::=  INTEGER {
        inappropriateAuthentication         (1),
        invalidCredentials                  (2),
        insufficientAccessRights            (3),
        invalidSignature                    (4),
        protectionRequired                  (5),
        noInformation                       (6),
        blockedCredentials                  (7),
        invalidQOPMatch                     (8),
        spkmError                           (9) }

serviceError  ERROR  ::=  {
        PARAMETER          OPTIONALLY-PROTECTED {
                SET {
                        problem             [0]     ServiceProblem,
                        COMPONENTS OF               CommonResults } }
        CODE                    id-errcode-serviceError }

ServiceProblem  ::=  INTEGER {
        busy                                (1),
        unavailable                         (2),
        unwillingToPerform                  (3),
        chainingRequired                    (4),
        unableToProceed                     (5),
        invalidReference                    (6),
        timeLimitExceeded                   (7),
        administrativeLimitExceeded         (8),
        loopDetected                        (9),
        unavailableCriticalExtension        (10),
        outOfScope                          (11),
        ditError                            (12),
        invalidQueryReference               (13),
        requestedServiceNotAvailable        (14),
        unsupportedMatchingUse              (15),
        ambiguousKeyAttributes              (16),
        saslBindInProgress                  (17) }

updateError  ERROR  ::=  {
        PARAMETER          OPTIONALLY-PROTECTED {
                SET {
                        problem             [0]   UpdateProblem,
                        attributeInfo       [1]   SET SIZE (1..MAX) OF CHOICE {
                                attributeType               AttributeType,
                                attribute                   Attribute } OPTIONAL,
                        COMPONENTS OF       CommonResults } }
        CODE                        id-errcode-updateError }

UpdateProblem  ::=  INTEGER {
        namingViolation                         (1),
        objectClassViolation                    (2),
        notAllowedOnNonLeaf                     (3),
        notAllowedOnRDN                         (4),
        entryAlreadyExists                      (5),
        affectsMultipleDSAs                     (6),
        objectClassModificationProhibited       (7),
        noSuchSuperior                          (8),
        notAncestor                             (9),
        parentNotAncestor                       (10),
        hierarchyRuleViolation                  (11),
        familyRuleViolation                     (12) }
```

```
-- 属性类型 --

id-at-family-information        OBJECT IDENTIFIER    ::=    {id-at 64}

END -- 号码簿抽象服务
```

附 件 B

基本访问控制的操作语义

（本附件不是本建议书 | 国际标准的组成部分）

本附件包含许多个图表，用于描述与基本访问控制有关的语义，适用于号码簿操作的处理（见图 B.1-图 B.16）。

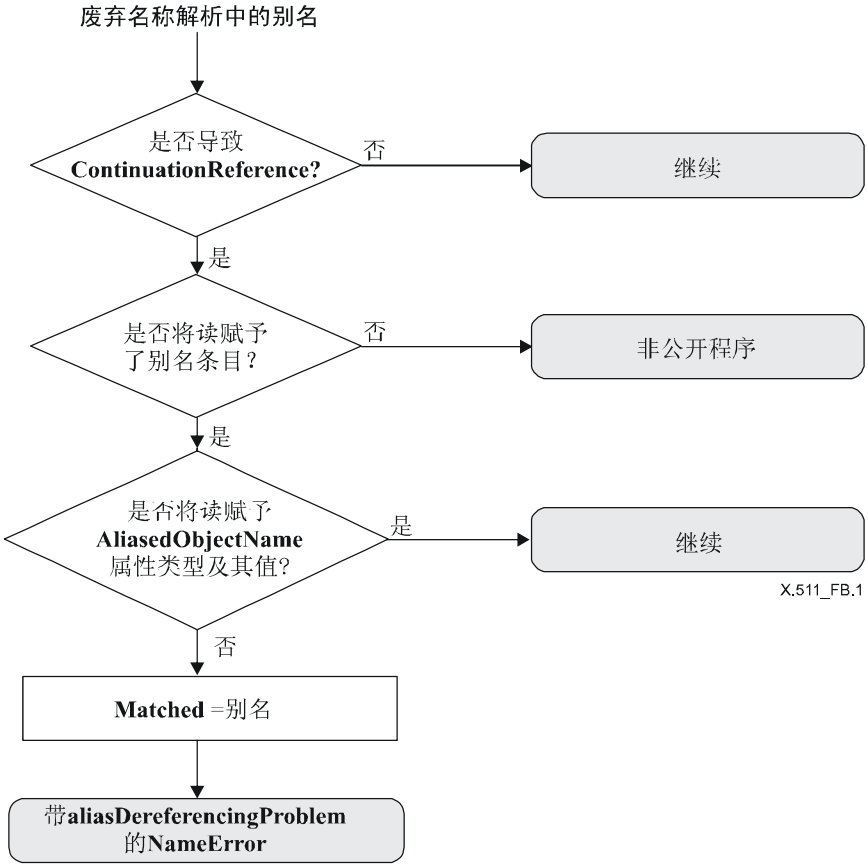

图 B.1－在名称解析中废弃别名

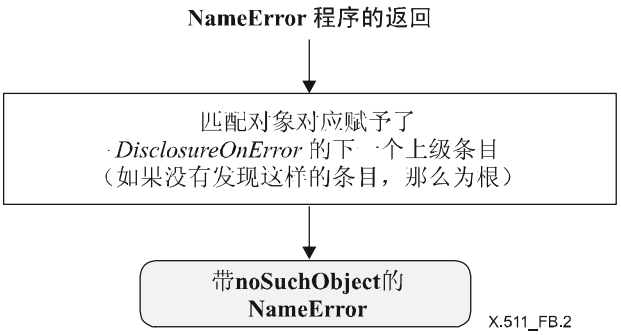

图 B.2－NameError 的返回

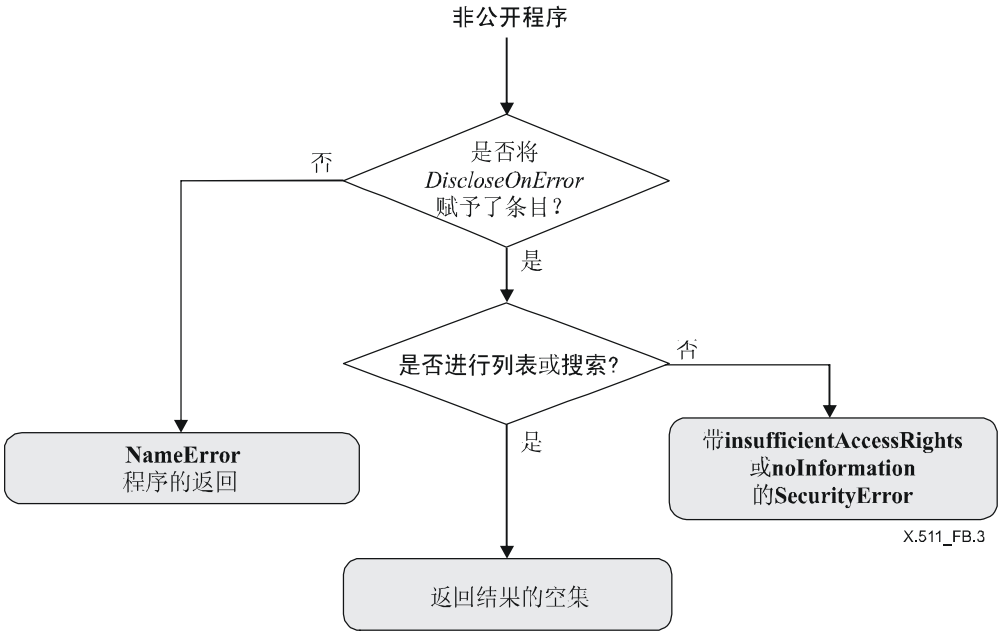

图 B.3－不透露条目是否存在

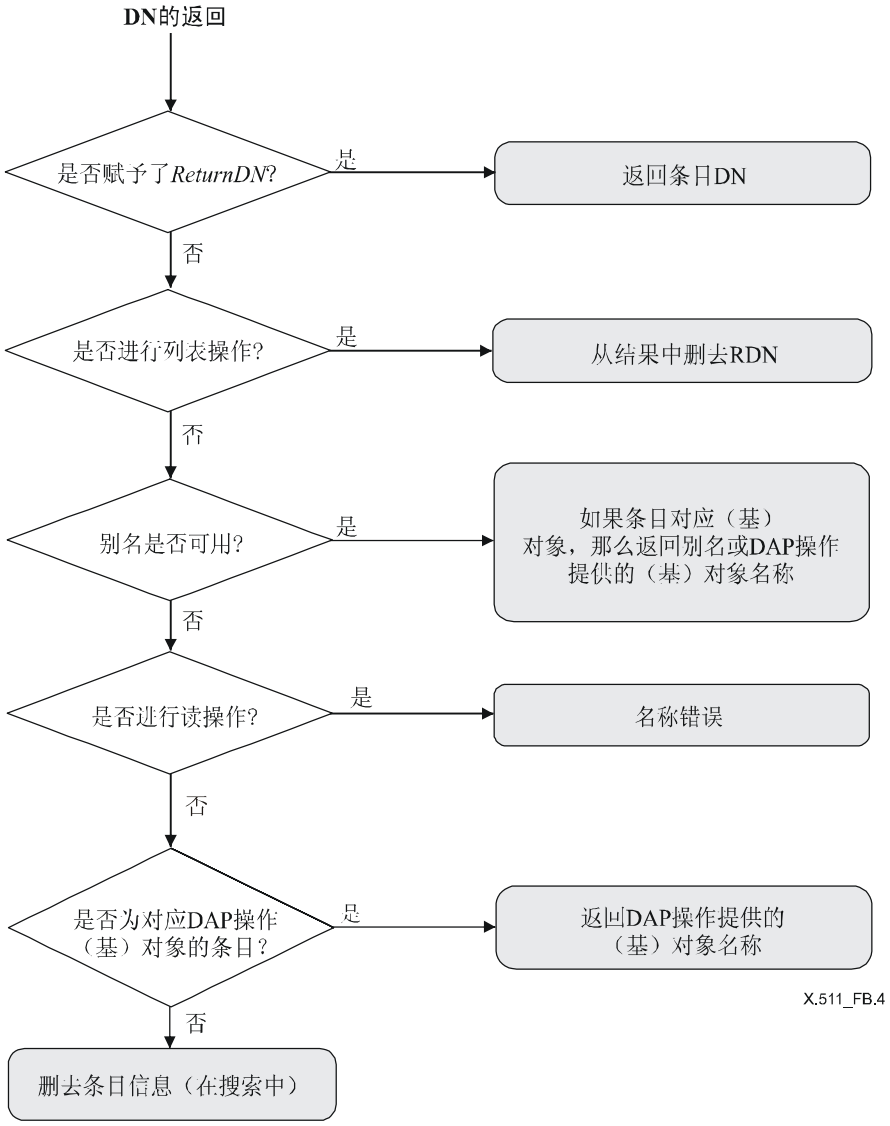
DN的返回
是否赋予了ReturnDN?
是
返回条目DN
否
是否进行列表操作?
是
从结果中删去RDN
否
别名是否可用?
是
如果条目对应（基）
对象，那么返回别名或DAP操作
提供的（基）对象名称
否
是否进行读操作?
是
名称错误
否
是否为对应DAP操作
（基）对象的条目?
是
返回DAP操作提供的
（基）对象名称
否
删去条目信息（在搜索中）
X.511_FB.4

图 B.4－不同名称的返回

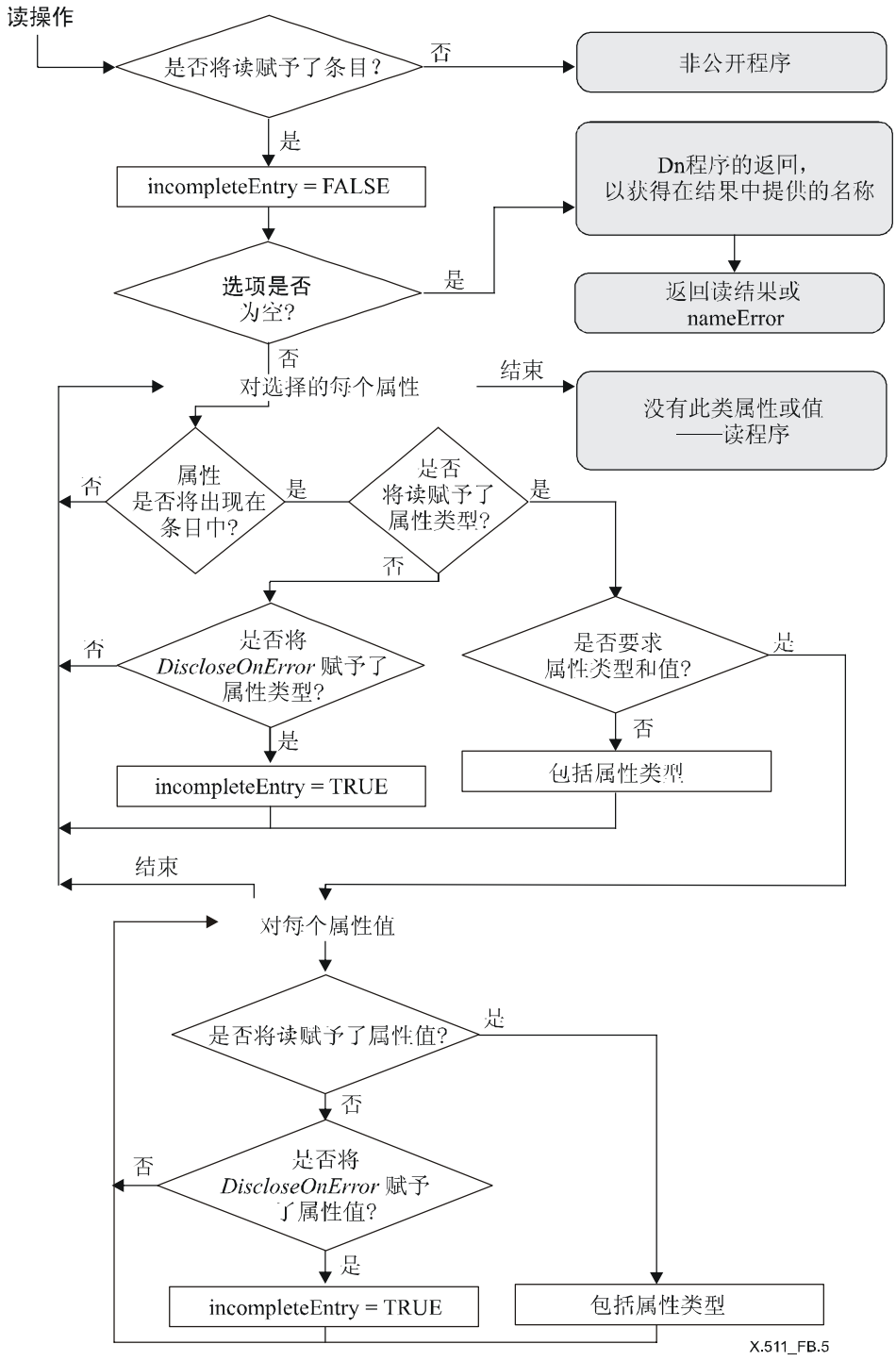
读操作
是否将读赋予了条目?
否
非公开程序
是
incompleteEntry = FALSE
Dn程序的返回,
以获得在结果中提供的名称
选项是否
为空?
是
返回读结果或
nameError
否
对选择的每个属性
结束
没有此类属性或值
——读程序
属性
是否将出现在
条目中?
否
是
是否
将读赋予了
属性类型?
是
否
是否将
DiscloseOnError 赋予了
属性类型?
否
是否要求
属性类型和值?
是
是
否
incompleteEntry = TRUE
包括属性类型
结束
对每个属性值
是否将读赋予了属性值?
是
否
是否将
DiscloseOnError 赋予
了属性值?
否
是
incompleteEntry = TRUE
包括属性类型
X.511_FB.5

图 B.5—读操作

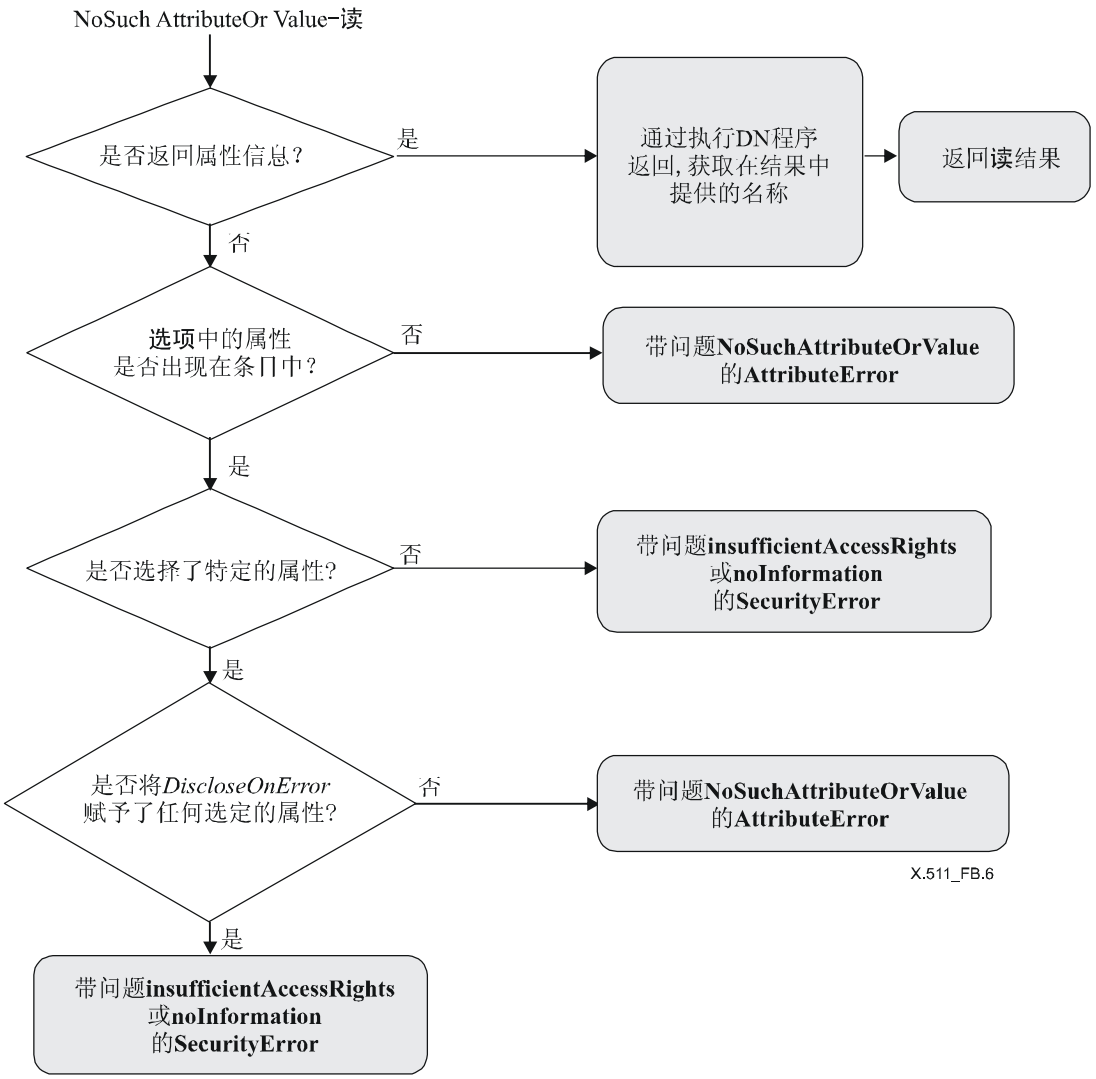

图 B.6－没有此类属性或值可读

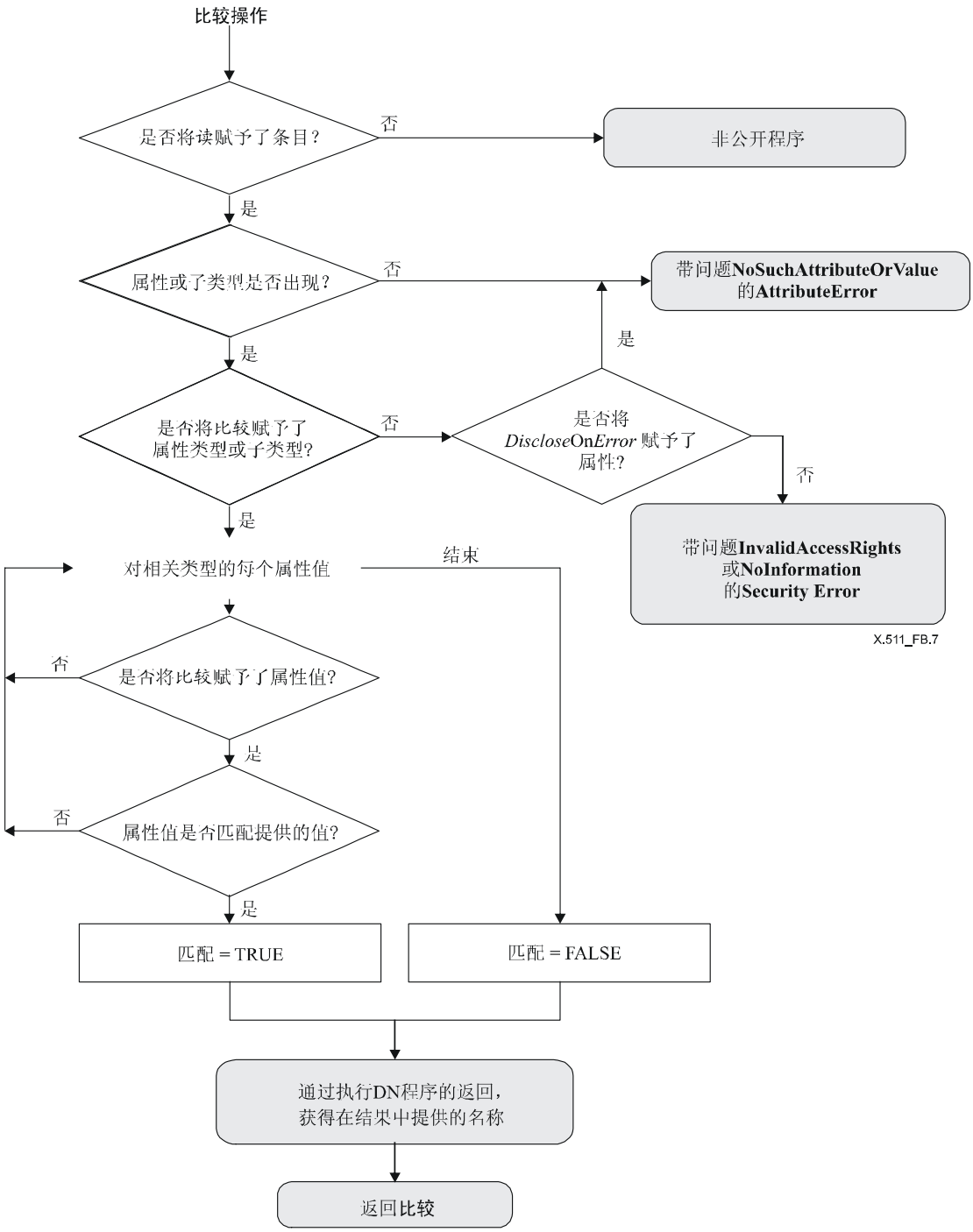
比较操作
是否将读赋予了条目?
否
非公开程序
是
属性或子类型是否出现?
否
带问题NoSuchAttributeOrValue
的AttributeError
是
是否将比较赋予了
属性类型或子类型?
否
是否将
DiscloseOnError 赋予了
属性?
是
否
带问题InvalidAccessRights
或NoInformation
的Security Error
X.511_FB.7
对相关类型的每个属性值
结束
否
是否将比较赋予了属性值?
是
否
属性值是否匹配提供的值?
是
匹配 = TRUE
匹配 = FALSE
通过执行DN程序的返回,
获得在结果中提供的名称
返回比较

图 B.7－比较操作

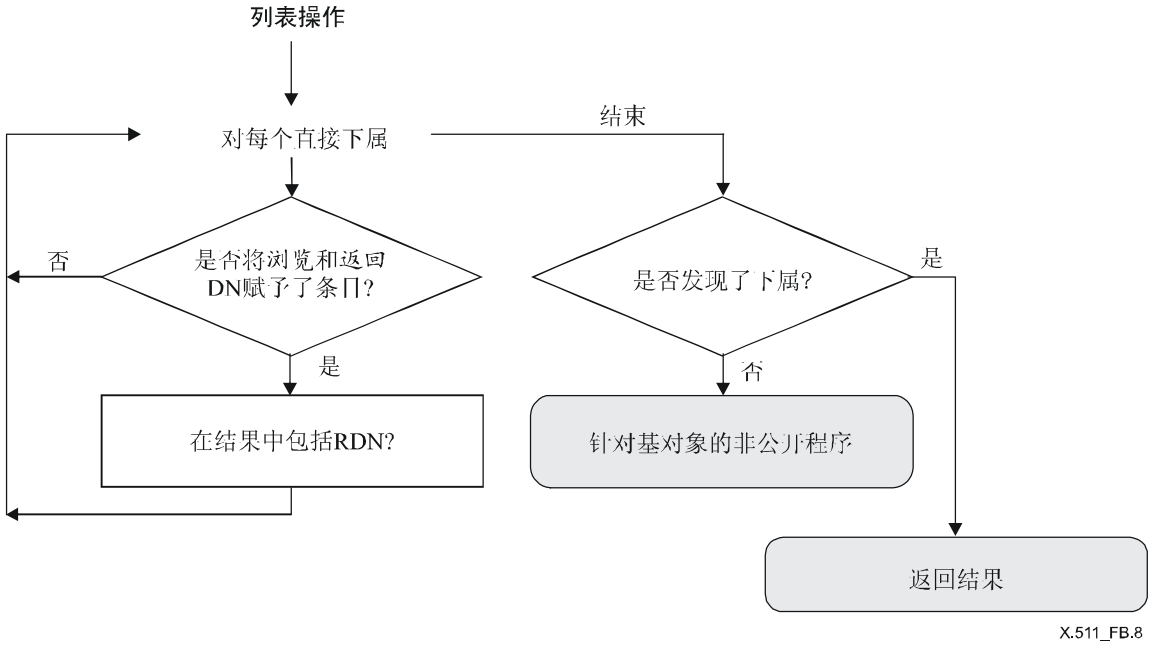
列表操作
对每个直接下属
结束
否
是否将浏览和返回
DN赋予了条目?
是
在结果中包括RDN?
是否发现了下属?
是
否
针对基对象的非公开程序
返回结果
X.511_FB.8

图 B.8－列表操作

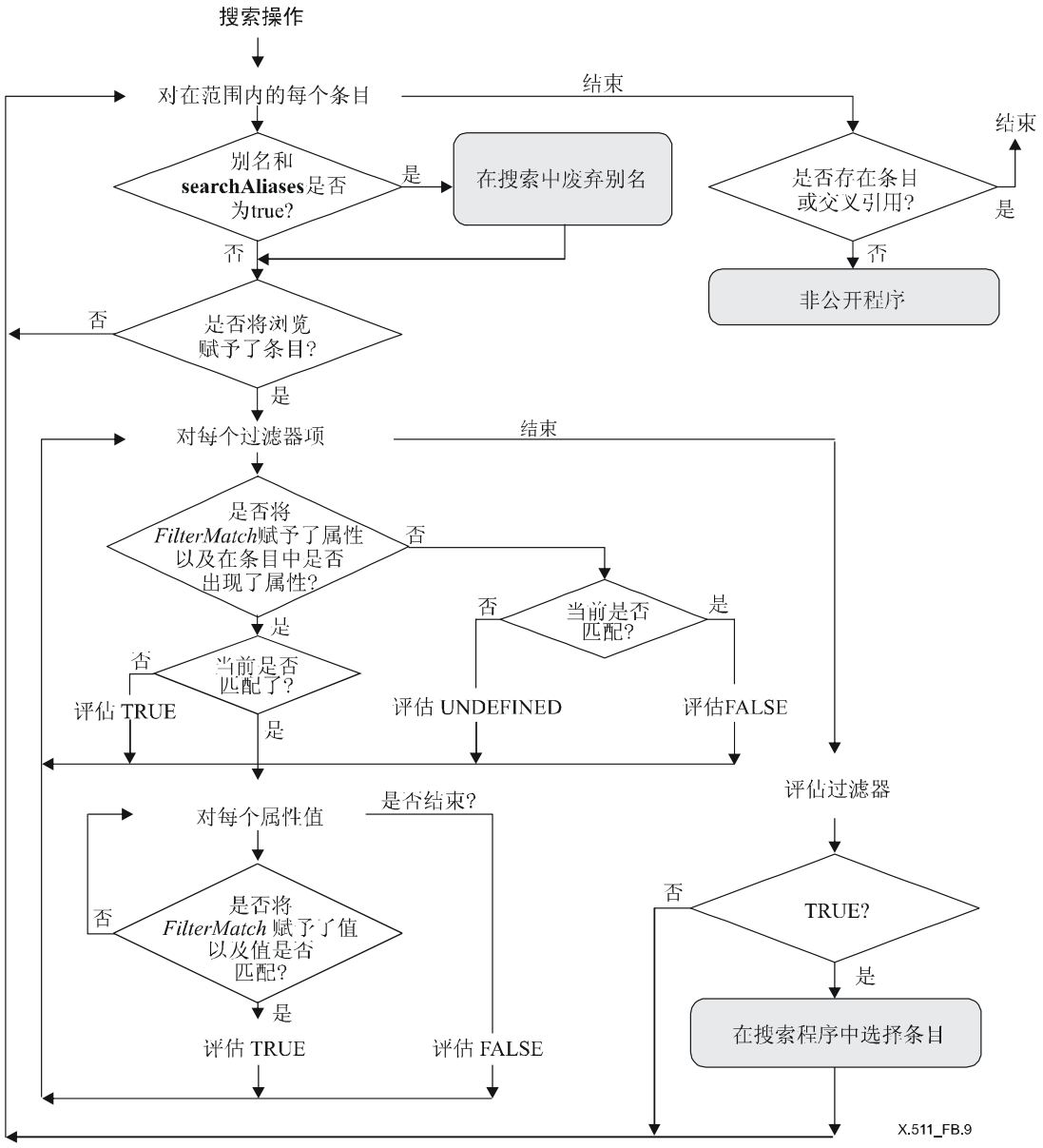
搜索操作
对在范围内的每个条目
结束
别名和
searchAliases是否
为true?
是
在搜索中废弃别名
否
是否存在条目
或交义引用?
结束
是
否
非公开程序
否
是否将浏览
赋予了条目?
是
对每个过滤器项
结束
是否将
FilterMatch赋予了属性
以及在条目中是否
出现了属性?
否
否
当前是否
匹配?
是
是
否
当前是否
匹配了?
评估 TRUE
评估 UNDEFINED
评估FALSE
是
评估过滤器
对每个属性值
是否结束?
否
是否将
FilterMatch 赋予了值
以及值是否
匹配?
否
TRUE?
是
是
在搜索程序中选择条目
评估 TRUE
评估 FALSE
X.511_FB.9

图 B.9－搜索操作

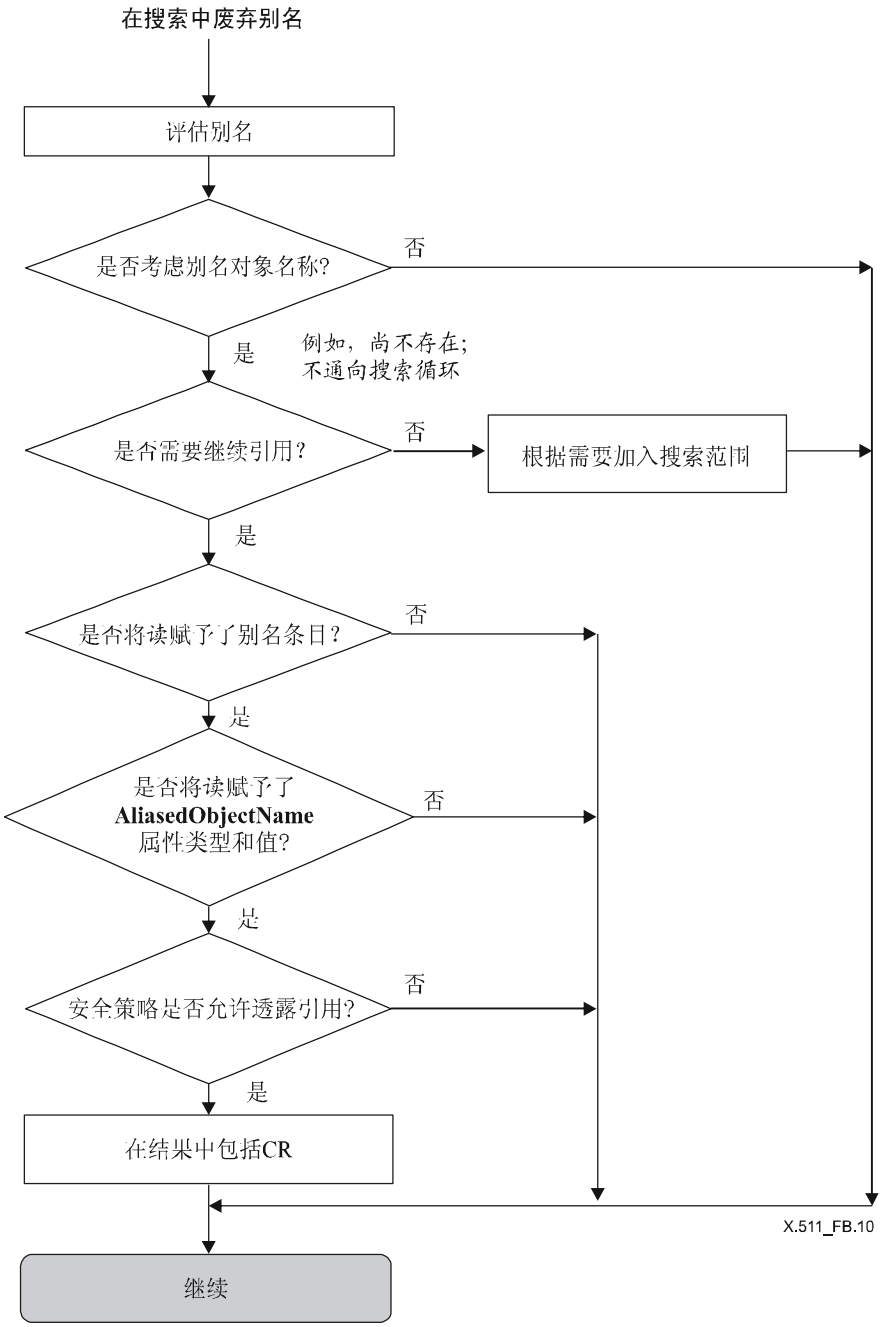

图 B.10－在搜索中废弃别名

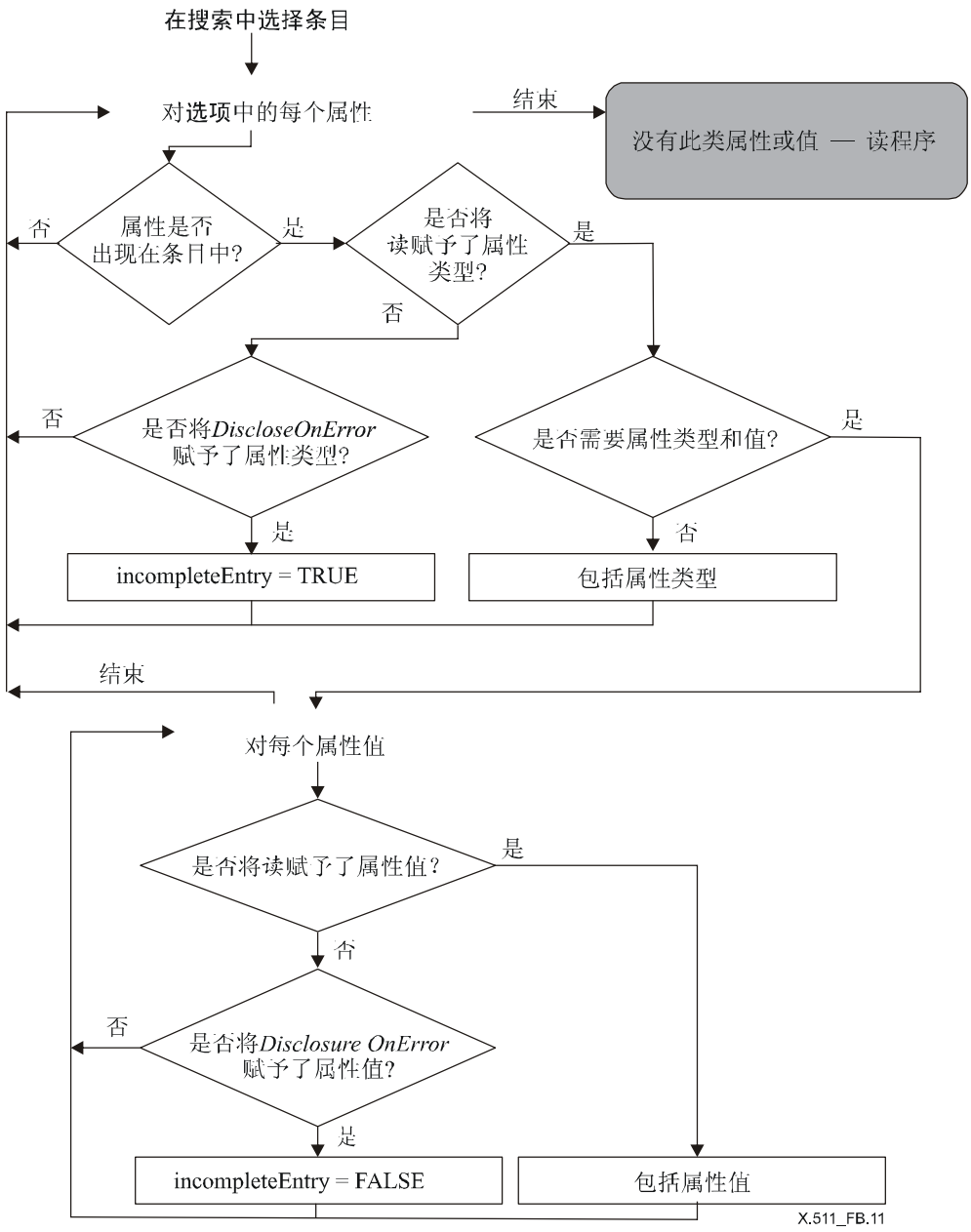

图 B.11－在搜索中选择条目

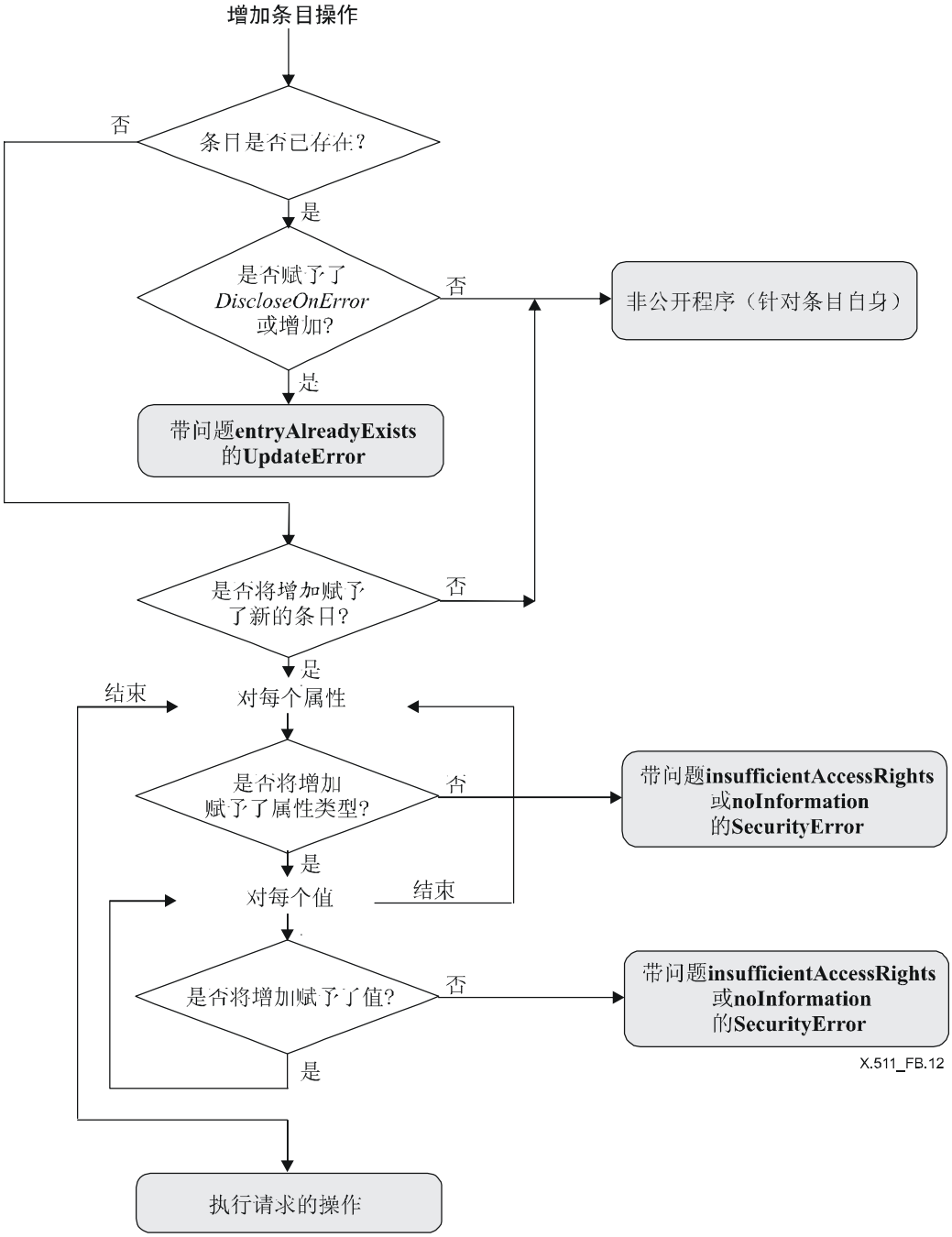

图 B.12－增加条目操作

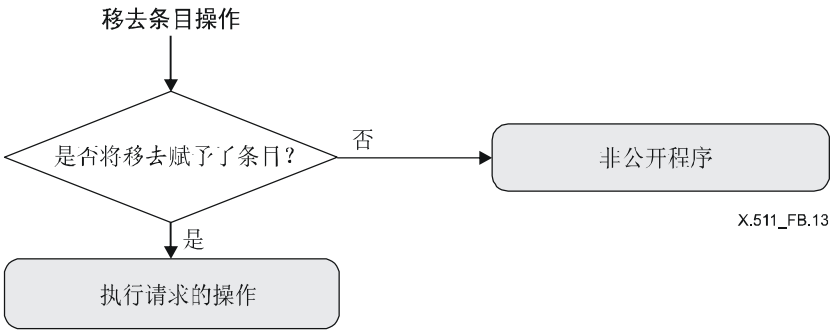

图 B.13－移去条目操作

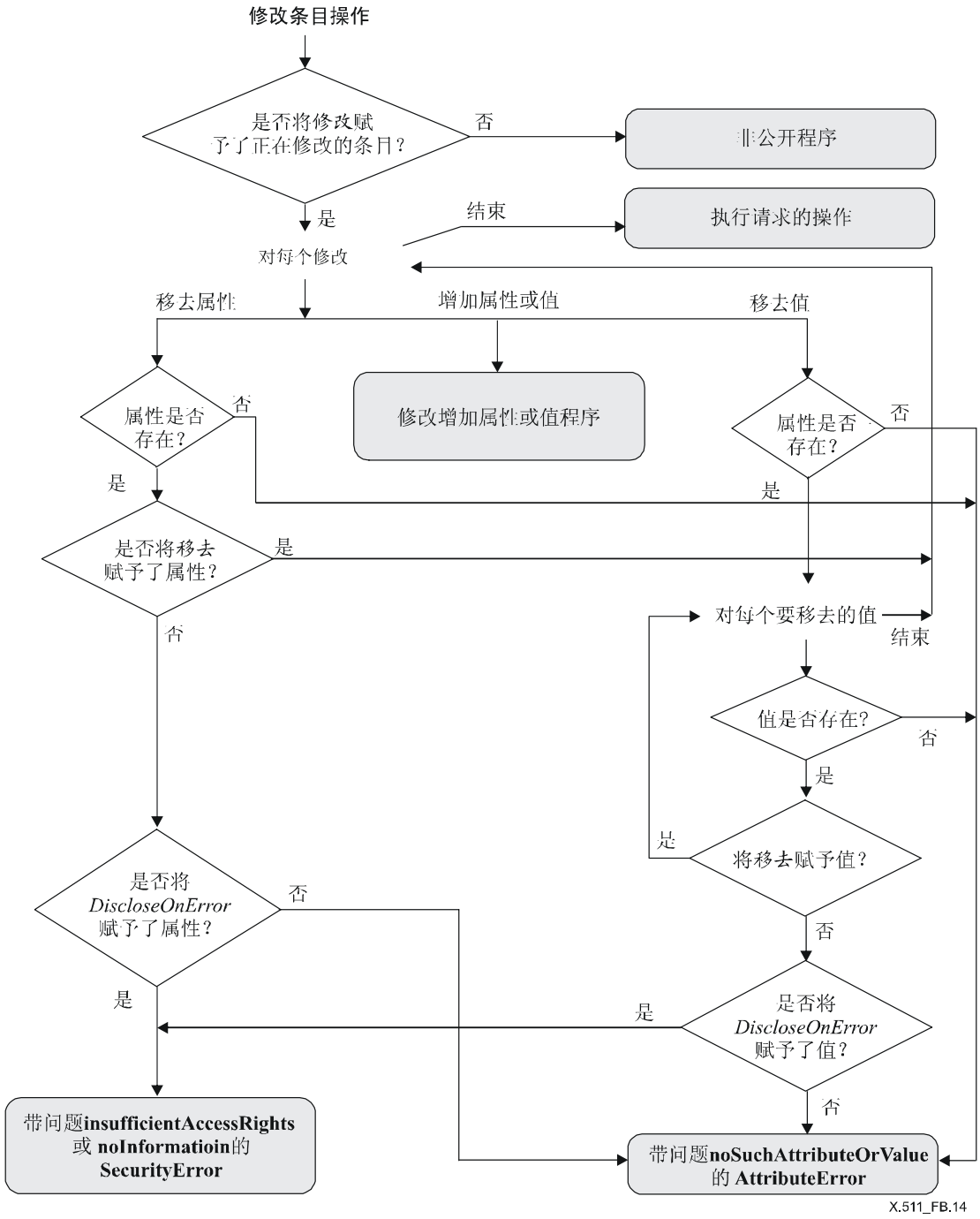
修改条目操作
是否将修改赋予了正在修改的条目？
否
非公开程序
是
结束
执行请求的操作
对每个修改
移去属性
增加属性或值
移去值
属性是否存在？
否
修改增加属性或值程序
属性是否存在？
否
是
是
是否将移去赋予了属性？
是
否
对每个要移去的值
结束
值是否存在?
否
是
是
将移去赋予值？
否
是否将DiscloseOnError赋予了属性？
否
是
是
是否将DiscloseOnError赋予了值？
否
带问题insufficientAccessRights或 noInformatioin的SecurityError
带问题noSuchAttributeOrValue的 AttributeError
X.511_FB.14

图 B.14－修改条目操作

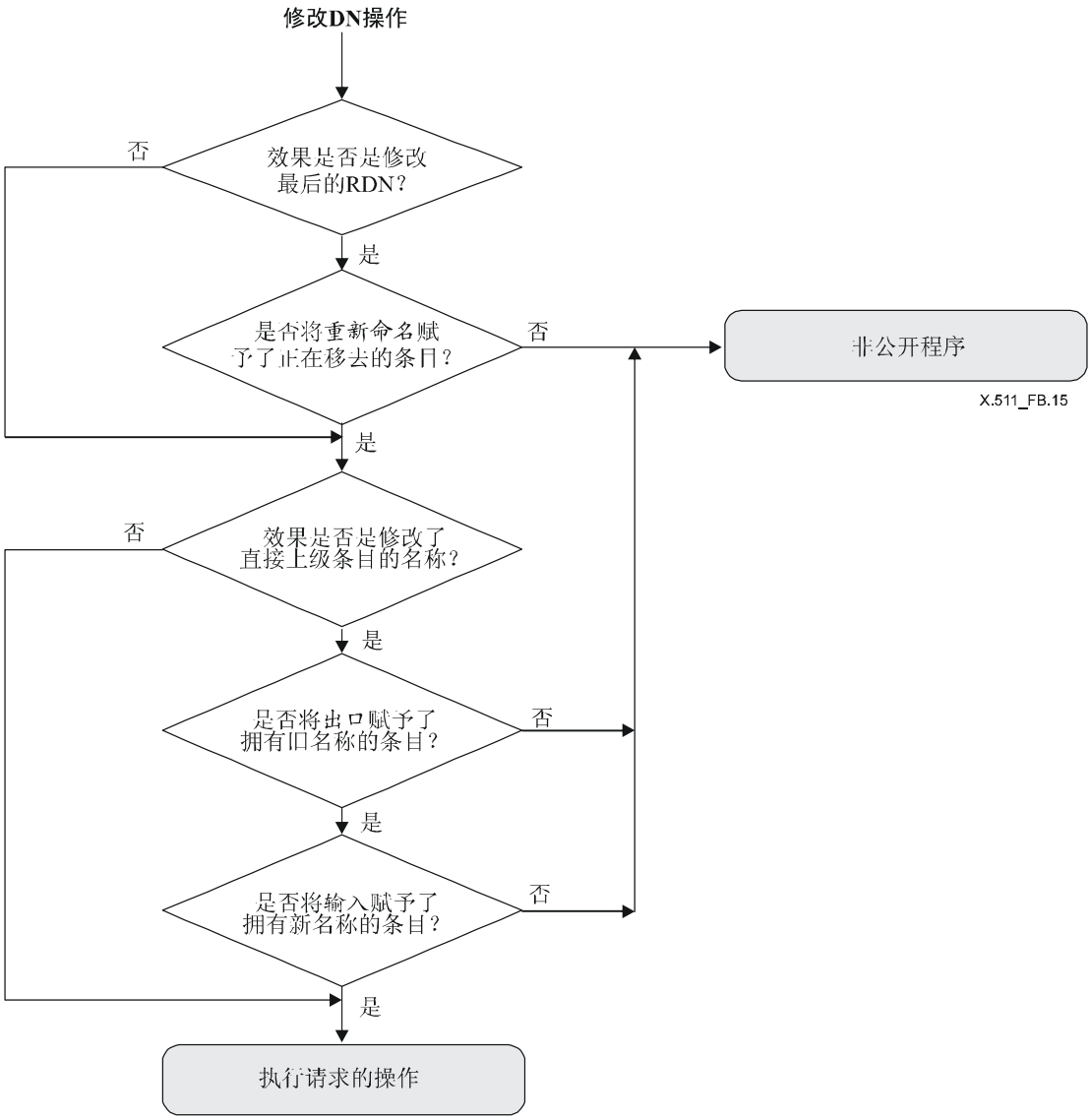
修改DN操作
效果是否是修改
最后的RDN?
否
是
是否将重新命名赋
予了正在移去的条目?
否
非公开程序
X.511_FB.15
是
效果是否是修改了
直接上级条目的名称?
否
是
是否将出口赋予了
拥有旧名称的条目?
否
是
是否将输入赋予了
拥有新名称的条目?
否
是
执行请求的操作

图 B.15－修改 DN 操作

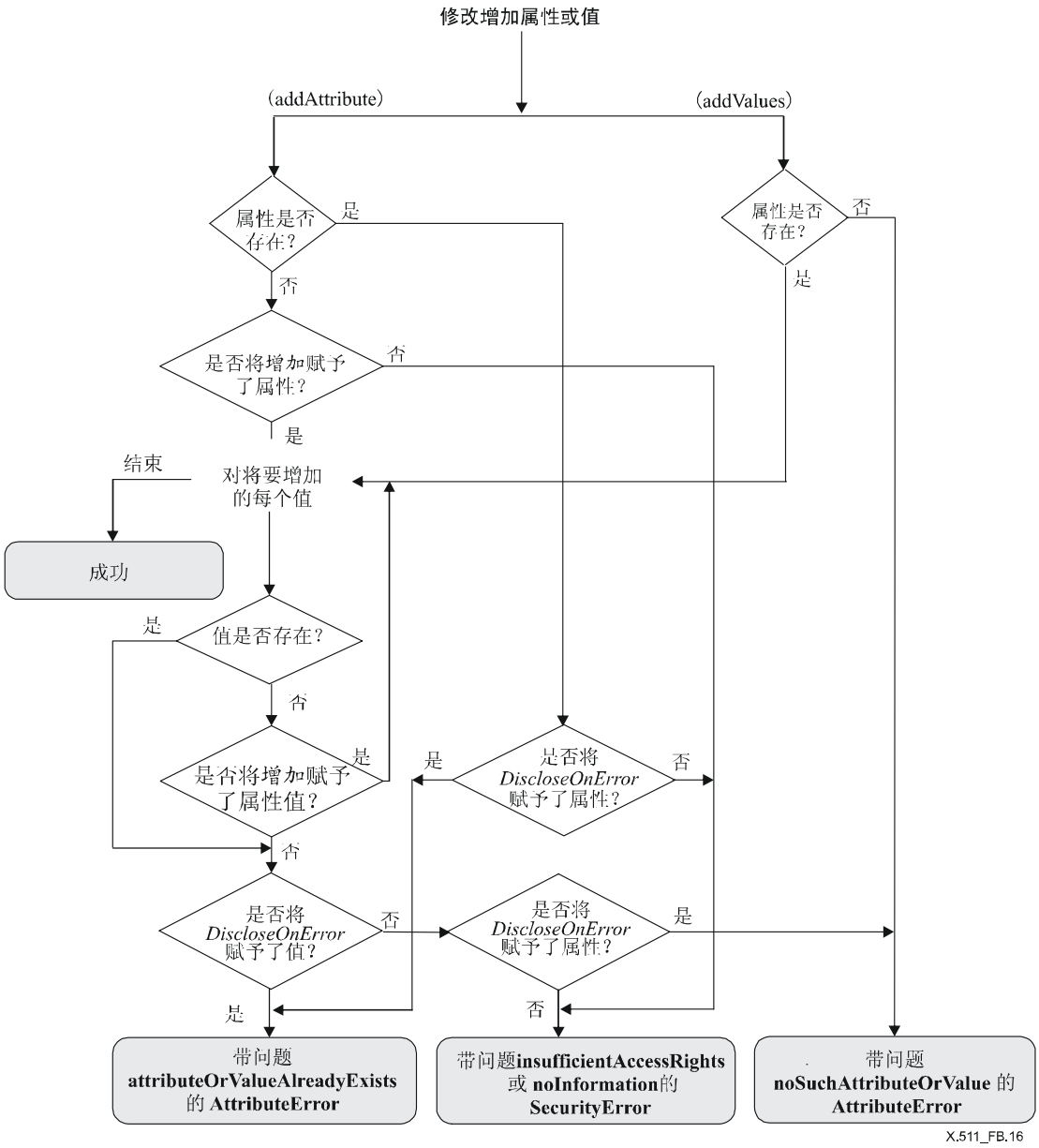

图 B.16－修改增加属性或值

附 件 C

搜索条目族举例

（本附件不是本建议书 | 国际标准的组成部分）

C.1 单个族举例

假设 Charles Smith 有多种通信模式：陆地电话、传真、移动电话和电子邮件，每种模式有其自身的相关参数。进一步假设 Charles Smith 有两个电子邮件账号，一个在其工作场所，一个在其家中，两个都提供了 POP3 邮箱和 SMTP 服务器。所有这些信息都可以存在于一个复合条目中，Charles Smith 的成员作为祖先，每种通信模式作为下属成员，每种电子邮件服务作为电子邮件通信模式的下属。这如下面的图 C.1 所示。由于作为祖先直接下属的所有成员都有相同的结构对象类别（**comAddr**），因此复合条目由一个单个的族组成。

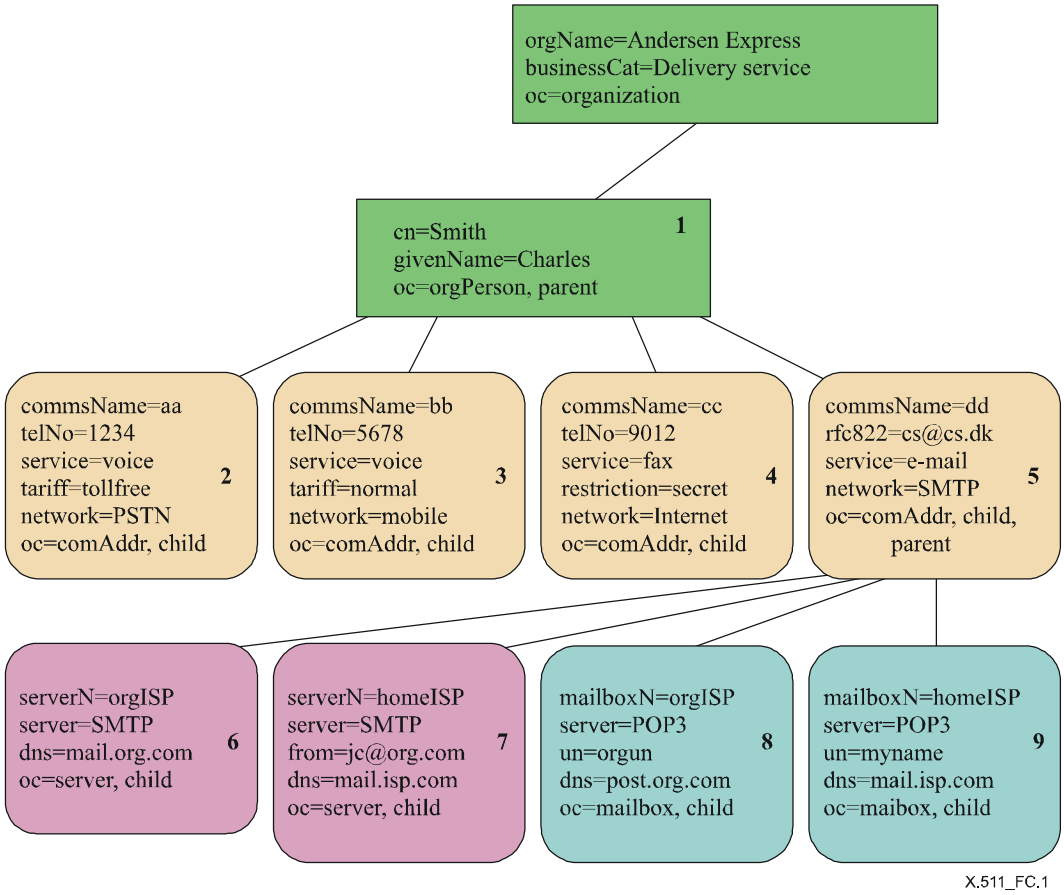

图 C.1－Charles Smith 的条目族

假设 **search** 请求由{...o=Andersen Express}的基对象、{telNo=1234 & tariff=normal}的过滤器、**wholeSubtree** 或 **oneLevel** 的子集产生。当 **familyGrouping** 参数设为以下值时：

a） **entryOnly**：族中将没有任何成员匹配于过滤器。

b） **strands** 或 **multiStrand**：族中将没有任何串或多串匹配于过滤器。

c） **compoundEntry**：成员 2 和成员 3 将一起匹配于过滤器，并将被标记为起作用的成员。所有的成员都将被标记为参与成员。

对上面情况 a）和情况 b），将不返回任何本复合条目中的内容。

对上面情况 c），返回的信息将依赖于族返回规范（如由 **EntryInformationSelection** 中的 **familyReturn** 给出）：

i） **contributingEntriesOnly**：各成员标记为起作用的成员，即成员 2 和成员 3 将被返回。

ii） **participatingEntriesOnly** 和 **compoundEntry**：复合条目中的所有成员都将被返回。

C.2 多个族举例

假设 Charles Smith 只有陆地电话和电子邮件，但还有两个具有相关参数的邮政地址。所有这些信息都可以存在于一个复合条目中，Charles Smith 的成员作为祖先，每种通信模式或每个邮政地址作为下属成员。这如下面的图 C.2 所示。由于作为祖先直接下属的所有成员属于两个不同的结构对象类别（**comAddr** 和 **postAddr**），因此复合条目由两个族组成，其中成员 1、成员 2 和成员 3 构成一个族，成员 1、成员 4、成员 5、成员 6、成员 7、成员 8 和成员 9 构成另一个族。

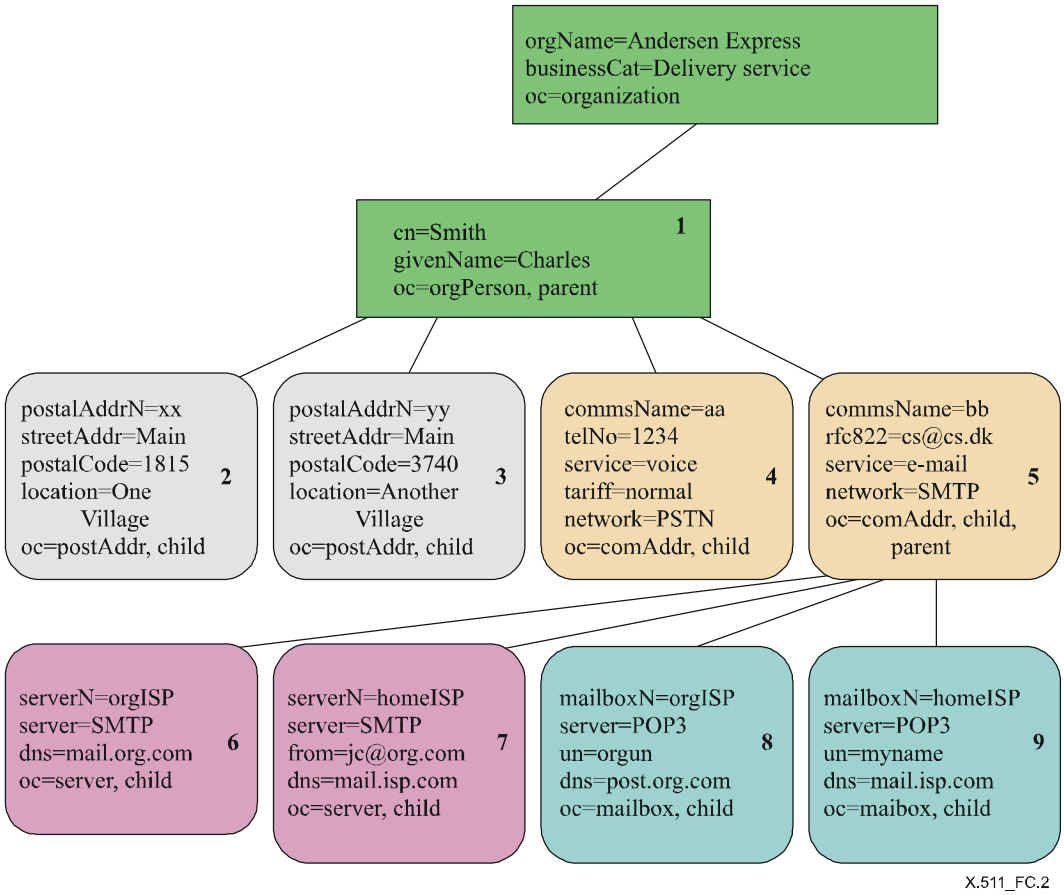

图 C.2－Charles Smith 条目族

C.2.1 过滤器举例1

现假设 Search 请求由{...o=Andersen Express}的基对象、{telNo=1234 & service=e-mail & streetAddr=Main & postalCode=3740}的过滤器、**wholeSubtree** 或 **oneLevel** 的子集产生。当 **familyGrouping** 参数设为以下值时：

a） **entryOnly**：复合条目中将没有任何单个成员匹配于过滤器。

b） **strands**：族中将没有任何单个串匹配于过滤器。

c） **multiStrand**：每个族中将没有任何串的组合或单个串匹配于过滤器。

d） **compoundEntry**：成员 2、成员 3、成员 4 和成员 5 将一起匹配于过滤器，并将被标记为起作用的成员。所有的成员都将被标记为参与成员。

对上面情况 a）、情况 b）和情况 c），将不返回任何本复合条目中的内容。

对上面情况 d），返回的信息将依赖于族返回规范：

i） **contributingEntriesOnly**：各成员标记为起作用的成员，即成员 2、成员 3、成员 4 和成员 5 将被返回。

ii） **participatingEntriesOnly** 和 **compoundEntry**：复合条目中的所有成员都将被返回。

C.2.2 过滤器举例2

如果将过滤器改为{rfc822=cs@cs.dk & service=e-mail & streetAddr=Main & postalCode=1815}，当 **familyGrouping** 参数设为以下值时：

a） **entryOnly**：复合条目中将没有任何单个成员匹配于过滤器。

b） **strands**：任何族中将没有任何单个串匹配于过滤器。

c） **multiStrand**：在成员 2 中结束的串以及任何通过成员 5 的串将匹配于过滤器。成员 2 和成员 5 有助于匹配，并将被标记为起作用的成员。成员 1、成员 2、成员 5、成员 6、成员 7、成员 8 和成员 9 将被标记为参与成员。

d） **compoundEntry**：成员 2 和成员 5 将一起匹配于过滤器，并将被标记为起作用的成员。所有的成员都将被标记为参与成员。

对上面情况 a）和情况 b），将不返回任何本复合条目中的内容。

对上面情况 c），返回的信息将依赖于族返回规范：

i） **contributingEntriesOnly**：各成员标记为起作用的成员，即成员 2 和成员 5 将被返回。

ii） **participatingEntriesOnly**：标记为参员成员的各成员将被返回，即成员 1、成员 2、成员 5、成员 6、成员 7、成员 8 和成员 9。

iii） **compoundEntry**：复合条目中的所有成员都将被返回。

对上面情况 d），返回的信息将依赖于族返回规范：

i） **contributingEntriesOnly**：各成员标记为起作用的成员，即成员 2 和成员 5 将被返回。

ii） **participatingEntriesOnly** 和 **compoundEntry**：复合条目中的所有成员都将被返回。

C.2.3 过滤器举例3

如果将过滤器改为{rfc822=cs@cs.dk & service=e-mail}，当 **familyGrouping** 参数设为以下值时：

a） **entryOnly**：只有成员 5 将匹配于过滤器，该成员将被标记为起作用的成员和参与成员。

b） **strands**：任何通过成员 5 的串将匹配于过滤器。成员 5 将被标记为起作用的成员。成员 1、成员 5、成员 6、成员 7、成员 8 和成员 9 将被标记为参与成员。

c） **multiStrand**：任何通过成员 5 的串以及任何邮政地址族的串都将匹配于过滤器。成员 5 将被标记为起作用的成员。成员 1、成员 2、成员 3、成员 5、成员 6、成员 7、成员 8 和成员 9 将被标记为参与成员。

d） **compoundEntry**：成员 5 将匹配于过滤器，并将被标记为起作用的成员。所有的成员都将被标记为参与成员。

对上面情况 a），返回的信息将依赖于族返回规范：

i） **contributingEntriesOnly** 和 **participatingEntriesOnly**：成员 5 将被返回。

ii） **compoundEntry**：复合条目中的所有成员都将被返回。

对上面情况 b），返回的信息将依赖于族返回规范：

i） **contributingEntriesOnly**：成员 5 将被返回。

ii） **participatingEntriesOnly**：标记为参与成员的所有成员都将被返回，即成员 1、成员 5、成员 6、成员 7、成员 8 和成员 9。

iii） **compoundEntry**：复合条目中的所有成员都将被返回。

对上面情况 c），返回的信息将依赖于族返回规范：

i) **contributingEntriesOnly**：成员 5 将被返回。

ii) **participatingEntriesOnly**：标记为参与成员的所有成员都将被返回，即成员 1、成员 2、成员 3、成员 5、成员 6、成员 7、成员 8 和成员 9。

iii) **compoundEntry**：复合条目中的所有成员都将被返回。

对上面情况 d），返回的信息将依赖于族返回规范：

i) **contributingEntriesOnly**：成员 5 将被返回。

ii) **participatingEntriesOnly** 和 **compoundEntry**：复合条目中的所有成员都将被返回。

C.2.4 过滤器举例4

如果将过滤器改为{cn=Smith & givenName=Charles}。只有祖先将匹配于过滤器。

a） **entryOnly**：只有祖先（成员 1）将被标记为起作用的成员和参与成员。

b） **strands**、**multiStrand** 和 **compoundEntry**：祖先将被标记为起作用的成员，所有的成员都将被标记为参与成员。

对上面情况 a），返回的信息将依赖于族返回规范：

i） **contributingEntriesOnly** 和 **participatingEntriesOnly**：成员 1 将被返回。

ii） **compoundEntry**：复合条目中的所有成员都将被返回。

对上面情况 b），返回的信息将依赖于族返回规范：

i） **contributingEntriesOnly**：成员 1 将被返回。

ii） **participatingEntriesOnly** 和 **compoundEntry**：复合条目中的所有成员都将被返回。

附 件 D

修正案和勘误表

（本附件不是本建议书 | 国际标准的组成部分）

本号码簿规范的这一版本包括如下对前一版本的修正案草案内容，该草案经 ISO/IEC 投票批准：

— 修正案 1：支持 DSP 分页结果的扩展；

— 修正案 2：支持友好属性概念的扩展；

— 修正案 3：尽可能实现 X.500 与 LDAP 之间的结合。

本号码簿规范的这一版本不包括任何技术上的勘误，用于纠正以下有缺陷的报告：308、309、313 和 316。

ITU-T 系列建议书

A系列	ITU-T工作的组织
D系列	一般资费原则
E系列	综合网络运行、电话业务、业务运行和人为因素
F系列	非话电信业务
G系列	传输系统和媒质、数字系统和网络
H系列	视听及多媒体系统
I系列	综合业务数字网
J系列	有线网络和电视、声音节目及其它多媒体信号的传输
K系列	干扰的防护
L系列	电缆和外部设备其它组件的结构、安装和保护
M系列	电信管理，包括TMN和网络维护
N系列	维护：国际声音节目和电视传输电路
O系列	测量设备的技术规范
P系列	电话传输质量、电话设施及本地线路网络
Q系列	交换和信令
R系列	电报传输
S系列	电报业务终端设备
T系列	远程信息处理业务的终端设备
U系列	电报交换
V系列	电话网上的数据通信
X系列	**数据网、开放系统通信和安全性**
Y系列	全球信息基础设施、互联网协议问题和下一代网络
Z系列	用于电信系统的语言和一般软件问题

瑞士印刷
2006年，日内瓦