

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利说明书

专利号 ZL 02805454.7

[45] 授权公告日 2010 年 1 月 20 日

[11] 授权公告号 CN 100583087C

[22] 申请日 2002.2.19 [21] 申请号 02805454.7
[30] 优先权
[32] 2001.2.26 [33] GB [31] 0104823.0
[86] 国际申请 PCT/GB2002/000706 2002.2.19
[87] 国际公布 WO2002/069185 英 2002.9.6
[85] 进入国家阶段日期 2003.8.25
[73] 专利权人 考珀瑞耶有限公司
地址 英国威尔特郡
[72] 发明人 D·G·保利
[56] 参考文献
CN1209208A 1999.2.24
CN1271442A 2000.10.25
EP0522363A2 1993.1.13
US5852822A 1998.12.22
审查员 刘宇儒

[74] 专利代理机构 上海专利商标事务所有限公司
代理人 张政权

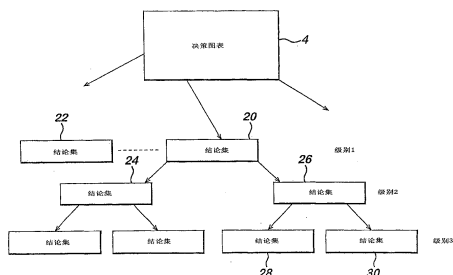
权利要求书 1 页 说明书 7 页 附图 3 页

[54] 发明名称

在数据库中组织数据

[57] 摘要

提供了数据库，其中结论集 (conclusion sets) (20, 22, 24, 26, 28 和 30) 被分为一系列分等级的级别 (级别 1, 级别 2, 级别 3)。数据被添加到第一级别 (级别 1) 的结论集，直到该结论集满为止。然后数据从结论集 (20) 移动到其次级结论集 (24, 26)，因此由于以相对少的盘存取操作移动多个项目，从而减少了将数据添加到数据库所选的盘存取量。



1. 一种在计算机系统中组织数据库(2)中的数据存储的方法,其特征在在于,在大容量存储媒体中以分等级的结构排列用于所述数据库的结论集(20, 24, 26, 28, 30, 40, 50, 60),各结论集排列成使得项目被插入到第一级别重要性的所选定的结论集,直到项数达到该所选定的结论集的阈值为止,然后将该所选定的结论集的内容移动到次级结论集,从而清空该所选定的结论集,在从该所选定的结论集移动内容之后,可进行对该结论集的进一步插入操作,其中,各结论集分布在所述数据库的决策图中,所述决策图包括多个分支节点,在这些分支节点处搜索关键字与决策标准匹配,以便确定应该采取哪条决策路径通过所述决策图;其中,各结论集形成在所述多个分支节点的一部分(41, 46)而不是全部处;并且,没有形成结论集分支节点确定了在形成了结论集的分支节点之间扩展的决策路径。

2. 如权利要求1所述的方法,其特征在在于,在移动过程中,将所选定的结论集中的数据与搜索标准相比较,以便选择要将项目传送到哪个直接从属的结论集。

3. 如权利要求2所述的方法,其特征在在于,搜索标准实际上是所述决策图的扩展。

4. 如权利要求1至3中任一项所述的方法,其特征在在于,一旦第N级别中的给定结论集中具有预定数量的项目,其内容就被移动到直接从属于它的结论集,这里N是从1到(L-1)的整数,其中所述分等级的结构包括L个级别。

5. 如权利要求1所述的方法,其特征在在于,具有在预定值上的级别重要性的结论集在使用中存储在快速存储器内。

6. 如权利要求1所述的方法,其特征在在于,构造所述决策图,使得在所述决策图的大部分中维持规定的结论集间距离。

7. 如权利要求6所述的方法,其特征在在于,在从结论集移动数据的过程中,根据特定的数据内容将所述数据移动到次级结论集。

8. 如权利要求1、6或7所述的方法,其特征在在于,结论集的内容被全体移动。

在数据库中组织数据

技术领域

本发明涉及在数据库中组织数据的方法，以及实现这样方法的数据库。

背景技术

典型的数据库，无论它们是在本申请人的共同待批准的英国专利申请 GB 0029238.3 中说明的类型，还是诸如“B-树”结构这样的其它已知的类型，它们都具有决策图或其它索引，指向保存匹配搜索标准的数据的结论集。此外和/或作为替代，结论集可保存指向匹配搜索关键字(search key)的数据的位置的指针。

在任何适当大小的数据库中，结论集被保存在大容量存储器媒体中，这里通常是指硬盘驱动器。硬盘装置（的速度）要比半导体存储器慢得多，并且由于得对这些大容量存储装置执行这些输入/输出(I/O)操作，所以数据库性能可能会受到影响。即使使用最小索引开销，通常数据库得执行 2 个 I/O 操作，作为将数据插入到索引和结论集中的读、修改和重写周期的一部分。

发明内容

根据本发明的第一方面，提供了在数据库中组织数据存储的方法，其中用于数据库的结论集被以分等级的结构排列，并且在其中这样安排该结论集，以使项(items)被插入到第一重要性级别的所选择的结论集，直到项的数量达到该所选择的结论集的阈值为止，然后该所选择的结论集的内容被移动到次级结论集，从而清空了所选择的结论集。

这样就可能提供修改的结论集结构，显著地减少了紧跟在决策图的输出之后的结论集的数量。

进一步，也可能在决策图中分布结论集。

现有技术领域的数据库具有可从决策图访问的单“层”结论集。插入到结论集中的每个新数据项将至少需要 2 个 I/O 操作来将该数据包含在内（条件是该数据可仅仅属于一个结论集，若插入的数据可属于不只一个结论集，则可能

有更多的 I/O 操作)。

通过以分等级的结构组织结论集,可大大减少可直接从决策图访问的结论集的数量。当然,就可能在诸如半导体存储器之类的快速存储器中保持最高等级重要性(即最高级别)的结论集。在本发明的那些实施例中,这一点尤其真实,其中通过决策图来分布结论集。

通过在半导体存储器中保持最高级别的结论集,当将数据插入到数据库中时,没有引起 I/O 消耗。这样就可能在关键字和数据插入操作过程中提供数据库性能的显著改善。

有利的是,高级别的结论集有效地高速缓存数据,直到该结论集变满或其中的项目的数量超过了预定级别为止。然后,通过将该结论集的内容移动到次级结论集来清空该结论集。在移动过程中,根据搜索标准(即搜索关键字)将数据分类,这样可期望数据随机分布在下一级(直接从属的)结论集之间。可在结论集结构中对多个分等级的级别重复这种填充(filling)和移动过程。

数据的移动可能需要(实际上经常需要)在一个或多个保持在大容量媒体存储器中的结论集之间进行数据传递。这样就引起了读盘和写盘操作,但现在这些操作是对结论集整体而发生的,而不是对结论集中的每个单独项发生的,从而每项目的 I/O 损耗就被大大减少了。

有利的是,在关键字检索(key retrieval)或删除过程中,检查决策标准匹配该搜索关键字的恰当的的最高级别结论集和每个次级结论集,以便查看匹配数据是否存储在其中。这样,与现有技术数据库比较,数据库查询开销增加了,但在插入量大、但查询量较低的某些数据库结构中,这一点是可接受的。

在结论集被分布在整个决策图中的本发明那些实施例中,可约束结论集间距离,以防止结论集出现得太频繁。

有利的是,结论集距离参数由整数 Q 定义。 Q 可采用比 0 大或等于 0 的数。这样,例如,可创建数据库,其中:

$Q=0$: 可在每个分支节点处形成结论集。

$Q=1$: 可在每个其它分支节点处形成结论集。

$Q=2$: 可在每个第二分支节点处形成结论集。

依此类推。

可在整个决策图上维持这些规则,直到达到决策图的最后 Q 层为止。在这些层中,有关结论集之间的分等级距离的规则变得不能执行,并且因此不严格

应用这些规则。

根据本发明的第二方面，提供了一种数据库，其中以分等级的结构排列用于该数据库的结论集，而且其中将结论集排列成使得项目被插入第一重要性级别的所选择的结论集，直到其中的项目数达到阈值为止，然后将该所选择的结论集的内容移动到次级结论集中，从而清空所选择的结论集。

根据本发明的第三方面，提供用于使数据处理器依照本发明的第一方面来运行的计算机程序产品。

附图说明

将通过例子，参考附图，进一步说明本发明，其中：

图 1 示意性的阐明了具有以传统方式排列的结论集的数据库；

图 2 示意性的阐明了具有依照本发明排列的结论集的数据库；

图 3 示意性的阐明了具有在决策图中分布的结论集的数据库，并构成本发明的实施例。

具体实施方式

图 1 中所示的数据库具有索引 2，并包含决策图 4 和多个结论集 6、8、10、12 和 14。通过决策图中的一个且仅一个通路到达每个结论集。然而，每个结论集随后指向数据存储 16 中的相关项目。

决策图 4 包含多个决策节点，在这些决策节点处，搜索关键字与决策标准匹配，以便确定那条路径应该通过决策图。在决策图内的关键字的内部组织并不构成本发明的部分，从而不必在这里详细说明。然而，在决策图中可利用诸如 B-树索引这样的现有技术索引结构。

在图 1 中所示的排列中，所有的结论集 6、8、10、12 和 14 都有相同的重要性，这样没有结论集在等级上比任何其它的结论集更重要，并且确实可能有成百上千甚至成千上万的结论集。

在图 2 中所示的排列中，结论集以分等级的结构排列。此处例示出的排列中，有三个级别的结论集，其中级别 1 是等级上最重要的，而级别 3 是等级上最不重要的。这样，在这个排列中，级别 1 的结论集的数量仅仅是级别 3 的结论集的数量四分之一，并且在这个例子中，一个级别 1 的结论集标记了到 6 个其它结论集的项目。明显的，随着级别数量增加，对于最不important级别处的给

定数量的结论集，级别 1 结论集的数量就逐渐减少。

现在假设我们想要插入一个项目到数据库中。依照对该项目的插入关键字来导航该决策图，这与现有技术数据库的情况相同，以便发现该项目属于哪个结论集。在图 1 中阐明的数据库中，这将导致唯一识别一个结论集。然而，在本发明中，这导致识别若干级别 1 的结论集（为清楚起见，仅显示了两个：20 和 22）中的一个结论集 20。有利的是，级别 1 的结论集 20 和 22 也保存在快速存储器中，即快速大容量媒体存储装置或更好是半导体存储器，这样与将数据插入到传统数据库的结论集 6、8、10、12 和 14 中的一个中所需的时间相比，将数据插入到结论集 20 中的时间开销少。实际上，若级别 1 的结论集 20 和 22 被保持在半导体存储器中，则在向其中写时，就没有引起 I/O 损耗。

当越来越多数据被插入到数据库中时，结论集 20 开始填充。一旦结论集 20 中的项目数量达到预定的数量时，这对应于该结论集满了，结论集 20 中的项目就被移动到属于等级结构中的级别 2 的下级（直接从属的）结论集 24 和 26。

通过继续存在于决策图 4 中的导航规则，确定较低级别的结论集 22 和 24 中的哪一个从级别 1 的结论集 20 接收项目的决策。这样，例如，若决策图 4 具有基于以位数的升序排列的单独位值的规则，则用于将数据从最高级别的结论集移动到第二级别的结论集 24 和 26 的规则将使用搜索关键字中的下一位来决定哪个结论集 24 或 26 应该是每个数据项的接受者。在这个移动过程中，结论集 20 被清空了。

当结论集层次内的每个级别中的每个结论集装满时，从第 N 级别到第 N+1 级别的移动过程就发生在该级别处。这样，一旦结论集 26 变满，它就转而将其数据移动到第三级别层次上的其次级的结论集 28 和 30。在这个例子中，该第三级别是最低级别的层次，并且结论集 28 和 30 不能将其数据向下传到次级结论集。然而，若 4 个或更多级别的结论集被包括在这个分等级的结构中，则当结论集 28 和 30 变满时，结论集 28 和 30 实际上可将其数据移动到自己的次级结论集。可以预期，假设关键字是随机分布的，在移动过程中，结论集 20 中的一半项目将转到结论集 24，而另外一半项目将转到结论集 26。该过程在移动的每个级别处重复，这样以使所有的项目基本上均等地分布在较低级别的结论集中。

可在将数据从结论集 26 移动到结论集 28 和 30 的操作中阐明移动的 I/O 损耗。在这个过程中，必须从结论集 26 中读数据，这需要一次读盘。较低级别

的结论集 28 和 30 也必须从盘读取数据，这需要两次读取操作。然后，来自结论集 26 的数据被分类，以正确指向目的集 28 或 30，然后更新项目，接着用于两个较低级别的结论集 28 和 30 的数据被写回到盘，这需要两个写操作。然后结论集 26 被清空，这需要一个写操作。这样这就给出了每个移动 6 个输入/输出操作的全部消耗。然而，这个单次移动可能影响成百上千的项目。还应该注意的，从最高级别的结论集 20 的移动将会更少，因为它将通常驻留在存储器中，从而仅要执行两次到集 24 和 26 的读操作和到这些集 24 和 26 的写操作。

假设图 2 中所示的分等级结构，结论集具有 S 个子级别（在图 2 的情况下， $S=2$ ， $S=(L-1)$ ，其中 L 是级数），将项目移动到最底部级别所需要的 I/O 操作的数量是 $6S$ 。然而，由于每个移动操作移动包含 E 个项目的整个（满的）结论集，所以每项目的 I/O 损耗由下式给出：

$$I/O \text{ cost} = \frac{6S}{E}$$

因此，若 E 足够大和/或 S 足够小，则与图 1 中所示的现有技术方案（每通道 2 个 I/O 损耗）相比较，可显著减少 I/O 损耗。这样，对于保持 100 个项目的结论集以及结论集深度为 8，每次插入的 I/O 损耗为 $6 \times \frac{8}{100} = 0.48$ 。

应该注意的是，对固定索引大小而言，虽然减少 S 增加了插入吞吐量，但是，为了要实现的好处，它还增加了必需保持在存储器中的结论集的数量。

当查询特定关键字的索引时，必须查询在趋向最低级别重要性的结论集的单路径中的每个结论集。这样，若分等级结构包含 L 个级别，则为了发现所有匹配该关键字的结果，必须查询 L 个结论集，这是由于该关键字可能驻留在层次中的任何级别处。这样，该索引方案增加了吞吐量，或者增加了项目可被加到索引的容易程度，其代价是降低了查询性能。然而，在任何得接收大量的数据但不经常查询的应用中，这中折衷是可接受的。这样应用的例子是必须载入每笔交易、但仅查询那些涉及可疑活动的交易的欺诈检测系统。

这样，能通过将结论集的结构修改为分等级的结构，以提供改善的数据库性能。

发明者已经意识到不需要所有的结论集在数据库结构的最低级别处。如在上文中注意到的那样，在图 2 中所示的分布结构中，数据被插入到发现的第一结论集中，即级别 1 结论集，然后数据被逐渐移动到较低级别。

这样，如果结论集被分布在整个决策索引结构中，那么可进一步减少插入

开销。

图 3 示出了修改的数据库，其中，结论集以分等级的结构排列，且其中该结构扩展到决策图中。

图 3 阐明了决策图表的一部分，其中已经用结论集间距离 $Q=3$ 来构造该决策图，这样允许结论集 40、50 和 60 在每个第三分支节点处。假设我们在包含于数据库中的数据是使得结论集 40 刚刚被创建，但结论集 50 和 60 还没有被创建的数据的时候加入数据库。

在使用中，数据被添加到数据库，而且该数据的内容使得结论集 40 变满。一旦这个情况出现，就通过创建较低级别节点 42 或者节点 43，并将数据从结论集 40 分布到图 3 中的轮廓示出的新的临时结论集 44 和 45，从而一起重新分布结论集 40 内的信息。然后对数据库的填充继续进行，直到结论集 40 再次填满为止。然后，它试图将其内容重新分布到其次级结论集。这转而可使它们中的一个变满，并因此有必要将该满的临时结论集从节点 42 的出口路径处移除，并按需插入一个或多个进一步节点以及附加的结论集。

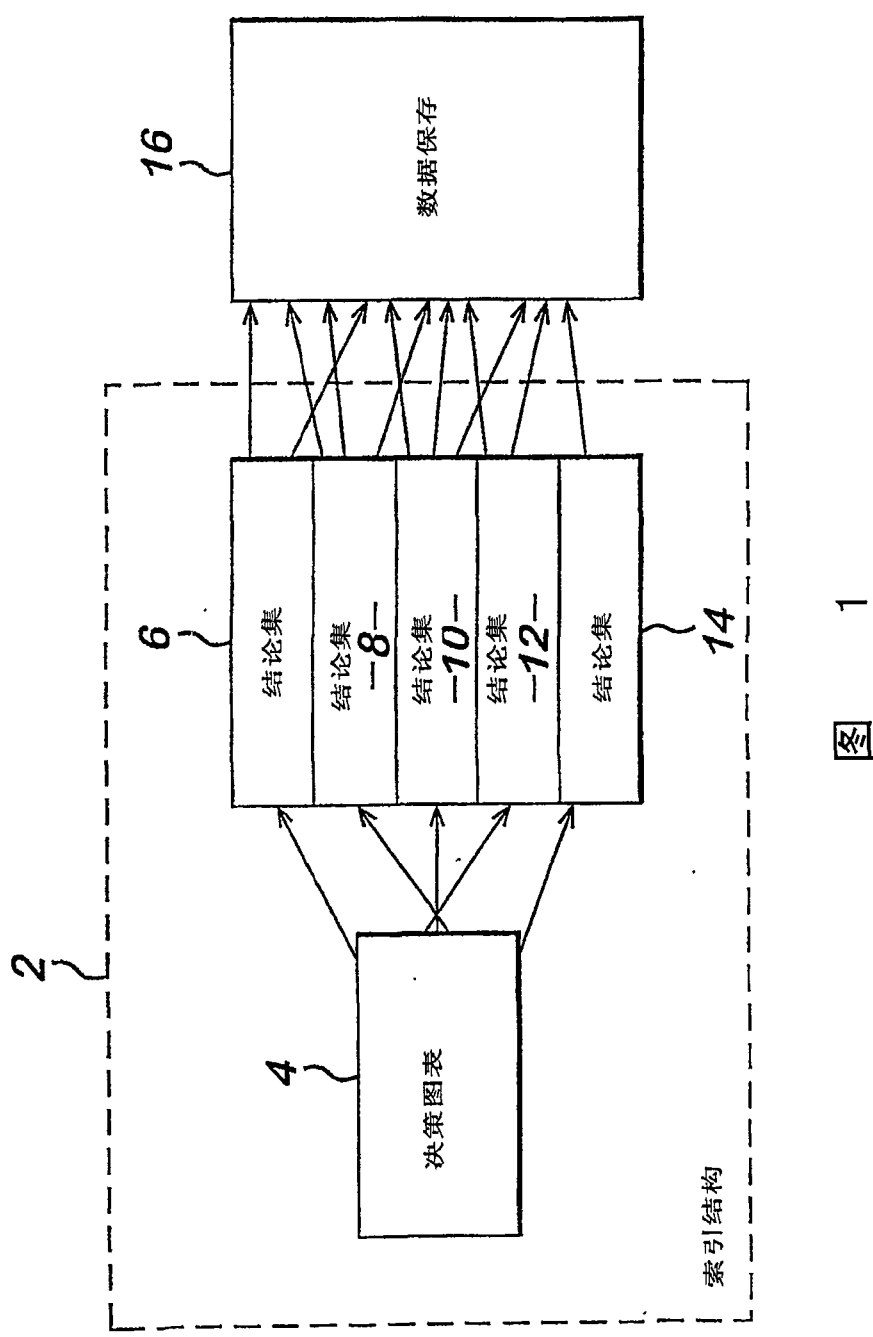
这可在建立结论集 50 作为节点 46 的出口这样的时间之前继续。此时，结论集 40 和 50 之间的结论集间距离于是对应于由数据库的设计者或用户指定的距离 Q 。在这些条件下，进行检查以查看是否有中间的结论集存在于节点 41 和 46 之间扩展的决策路径中。任何在这个决策路径中的结论集被移除，并且它们的内容被移动到结论集 50。

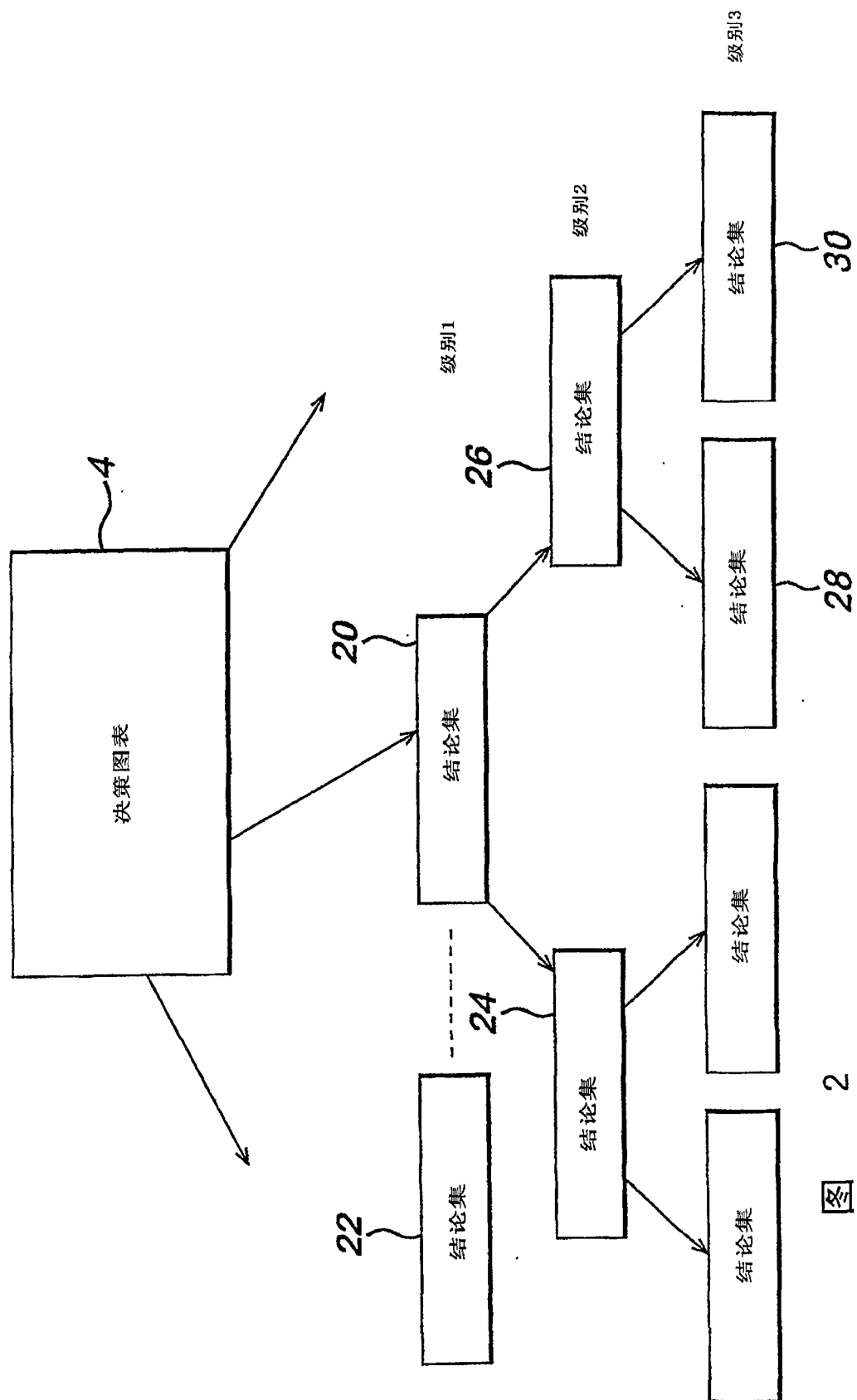
当结论集 50 填满时，可在创建结论集 60 的时间之前创建新的临时结论集，并依此类推。这样，不处于决策索引最低层中的结论集之间的结论集间距离就可能被保持在规定的结论集间间隔。在图 3 中所示的排列中，结论集 62 和 70 代表数据库的最低层，当然，结论集 60、62 和 70 可代表图 2 中所示排列中级别 1、2 和 3 的结论集。

如上所述，结论集项目向较低级别的结论集的大量重新分布导致与单独数据项的写（操作）相比较而言的减少的盘输入/输出困难(penalty)。进一步，由于项目被插入到发现的第一结论集中，所以图 3 中所示到结构中的数据项目消耗被进一步减少了。然后，满的结论集的每个项目被移动到通过遵循从满结论集通过要移动的条目的决策图而发现的合适的较低级别结论集。

当查询该数据库时，必需导航用于相关关键字的决策图，正好到达最低层级别的结论集，并且还必须查询在路径上发现的每个结论集，这是由于它可能

包含匹配搜索关键字标准的数据。





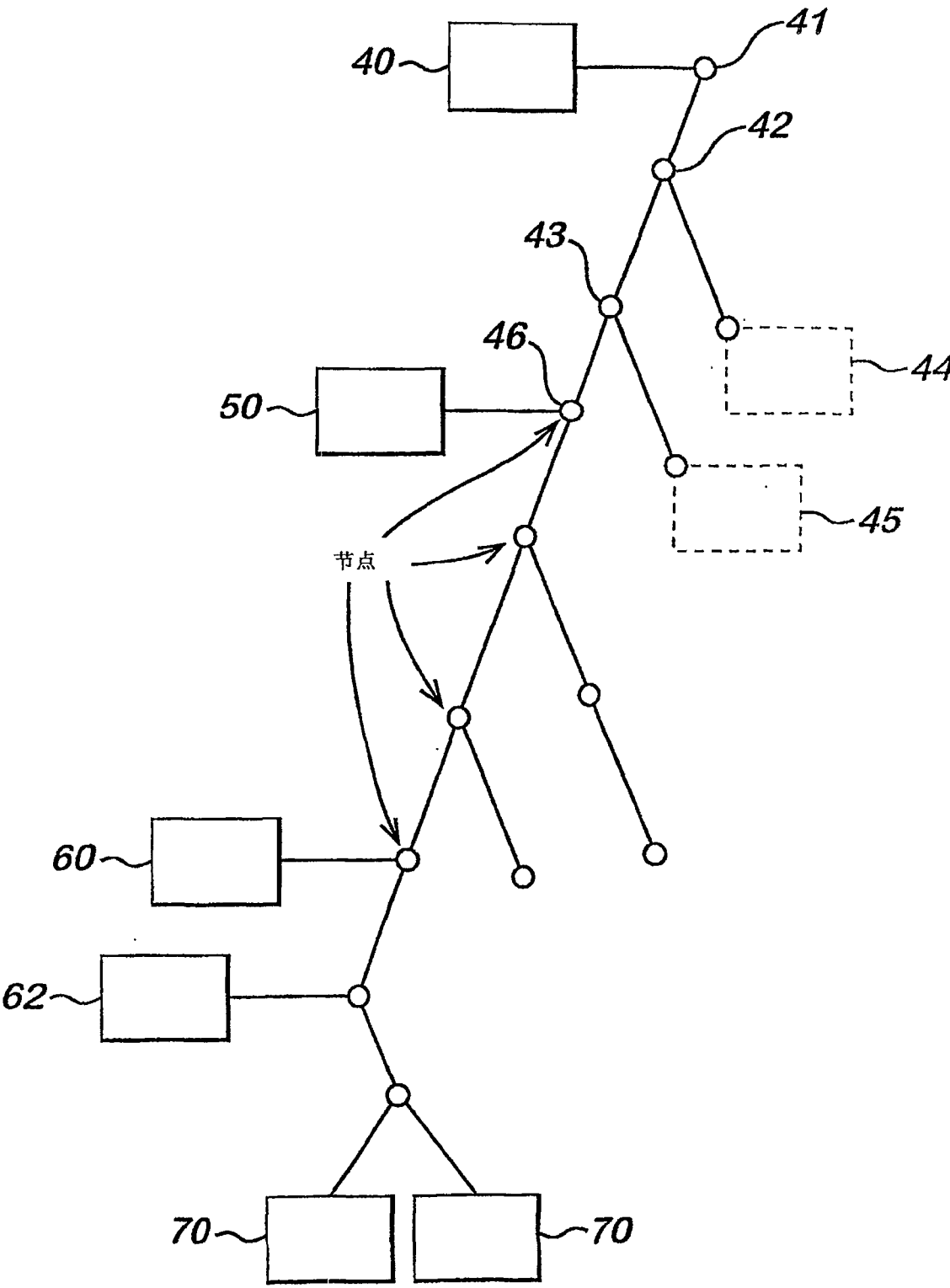


图 3