

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/00 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200480011372.9

[43] 公开日 2006 年 5 月 31 日

[11] 公开号 CN 1781094A

[22] 申请日 2004.2.27

[21] 申请号 200480011372.9

[30] 优先权

[32] 2003. 3. 10 [33] US [31] 10/385,421

[86] 国际申请 PCT/US2004/006044 2004.2.27

[87] 国际公布 WO2004/081711 英 2004.9.23

[85] 进入国家阶段日期 2005.10.27

[71] 申请人 尤尼西斯公司

地址 美国宾夕法尼亚

[72] 发明人 詹尼·坎贝尔·马扎加蒂

[74] 专利代理机构 中国国际贸易促进委员会专利商
标事务所
代理人 付建军

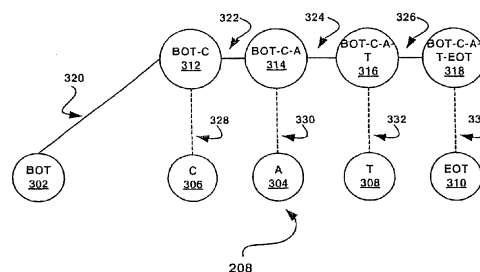
权利要求书 6 页 说明书 31 页 附图 18 页

[54] 发明名称

用于存储和访问互锁树数据仓库中的数据的
系统和方法

[57] 摘要

生成和/或访问包括互连的树的林所构成的基于树的数据仓库。基于树的数据仓库包括从第一根节点下垂的第一树，并可以包括多个树枝。第一树的每一个树枝都在叶节点中结束。每一个叶节点都可以代表最终产物，如下面比较全面地描述的。同一个基于树的数据仓库的第二根链接到代表最终产物的每一个叶节点。因此，第二根基本上是第一树的倒序的根，但第一树不重复。最后，基于树的数据仓库包括多个树，其中，这些树中的每一个树的根节点都是基本节点，如下面比较全面地描述的。这些树中的每一个树的基本根节点都可以链接到第一树的一个或多个树枝中的一个或多个节点。基于树的数据仓库的节点只包含指向基于树的数据仓库中的其他节点的指针。包括基于树的数据仓库的树林中的树的根可以链接到数据。



1. 一种用于生成基于树的数据仓库的系统，包括：
处理器；
5 连接到处理器的存储器；以及
用于创建基于树的数据仓库的至少一个级别的基于树的数据仓库生成器，所述基于树的数据仓库的至少一个级别包括第一树、第二树以及至少第三树，所述第一树中包括第一根和多个节点中的至少一个节点，所述第二树中包括第二根和第一树的至少一个节点，所述第三树中包括第三根和第一树的多个节点中的至少一个节点。
10
2. 根据权利要求 1 所述的系统，其中，数据仓库的节点包括至少一个基本节点、一个子组件节点和一个最终产物节点。
3. 根据权利要求 2 所述的系统，其中，基于树的数据仓库包括至少第一级别和第二级别，所述第一级别的最终产物包括第二级别的基本节点。
15
4. 根据权利要求 2 所述的系统，其中，基于树的数据仓库包括至少第一级别和第二级别，分解第一级别的最终产物，以创建第一级别的基本节点。
5. 根据权利要求 2 所述的系统，其中，基于树的数据仓库包括至少第一级别和第二级别，分解第二级别的最终产物，以创建第一级别的基本节点。
20
6. 根据权利要求 1 所述的系统，其中，节点包括指向基于树的数据仓库中的其他节点的指针。
7. 根据权利要求 1 所述的系统，其中，第一根和第二根包括指向基于树的数据仓库中的其他节点的指针，并包括非指针信息。
25
8. 根据权利要求 7 所述的系统，其中，第一根与开始指示符关联并以第一分层顺序访问基于树的数据仓库。
9. 根据权利要求 7 所述的系统，其中，第二根与结束指示符关联并以反转的第一分层顺序访问基于树的数据仓库。

10. 根据权利要求 7 所述的系统, 其中, 第三根包括与数据集的元素关联的非指针信息, 并基于数据集元素以第三分层顺序来访问基于树的数据仓库。

5 11. 根据权利要求 1 所述的系统, 进一步包括用于从基于树的数据仓库检索信息的基于树的数据仓库。

12. 根据权利要求 11 所述的系统, 其中, 基于树的数据仓库存取器进一步包括:

10 用于接收对要从基于树的数据仓库中检索的信息的请求的装置;
用于从基于树的数据仓库中检索请求的信息的装置; 以及
用于返回从基于树的数据仓库检索到的信息的装置。

13. 一种用于生成数据仓库的系统, 包括:

处理器;

连接到处理器的存储器; 以及

15 用于创建数据仓库的数据仓库生成器, 其中, 数据仓库包括:
被组织成多个连接的树的根节点和非根节点, 所述多个连接的树包括第一种类型的树和多个第二种类型的树的至少一个树, 所述第一种类型的树中包括了第一根和多个非根节点中的至少一个节点, 其中, 第二种树包括第二根节点和与第一种类型的树的节点共用的多个非根节点。

20 14. 根据权利要求 13 所述的系统, 其中, 数据仓库的节点包括至少一个基本节点、一个子组件节点和一个最终产物节点。

15. 根据权利要求 14 所述的系统, 其中, 第一种类型的树记录子组件节点与基本节点的至少一个组合的最终产物的顺序合成, 并提供对第一上下文中的数据仓库中的数据。

25 16. 根据权利要求 13 所述的系统, 其中, 多个第二种类型的树中的至少一个树提供对第二上下文中的数据仓库中的数据的访问。

17. 根据权利要求 14 所述的系统, 其中, 至少一个基本节点包括指向到节点的第二指针列表的第一 Null 指针, 第二 Null 指针, 第三 Null 指针和第四指针, 所述第二列表包括包含基本节点作为它

们的第二部分的节点。

18. 根据权利要求 14 所述的系统, 其中, 至少一个子组件节点包括指向第一节点的第一指针, 第一节点包括子组件节点的第一部分, 指向第二节点的第二指针, 第二节点包括子组件节点的第二部分, 指向到节点的第一指针列表的第三指针, 第一列表包括包含子组件作为它们的第一部分和第四 Null 指针的第一列表。

19. 根据权利要求 14 所述的系统, 其中, 至少一个最终产物节点包括至少指向第一部分的第一指针, 指向结束指示符第二部分的第二指针, 指向到节点的第二指针列表的第三 Null 指针和第四指针, 第二列表包括包含最终产物节点作为它们的第二部分的节点。

20. 根据权利要求 13 所述的系统, 其中, 代表开始指示符的根节点包括指向到节点的第一指针列表的第一 Null 指针, 第二 Null 指针, 第三指针, 第一列表包括包含开始指示符作为第一部分和第四 Null 指针的节点。

21. 根据权利要求 13 所述的系统, 其中, 代表数据集元素的根节点包括指向到节点的第二指针列表的第一 Null 指针, 第二 Null 指针, 第三 Null 指针和第四指针, 第二列表包括包含数据集元素作为第二部分的节点。

22. 根据权利要求 13 所述的系统, 其中, 代表最终产物的根节点包括指向到节点的第二指针列表的第一 Null 指针, 第二 Null 指针, 第三 Null 指针和第四指针, 第二列表包括包含最终产物作为第二部分的第二级别的节点。

23. 根据权利要求 13 所述的系统, 进一步, 包括用于访问存储在互锁树数据仓库中的信息的存取器。

24. 根据权利要求 23 所述的系统, 其中, 存取器进一步包括: 用于接收对要从基于树的数据仓库中检索的信息的请求的装置; 用于从基于树的数据仓库中检索请求的信息的装置; 以及用于返回从基于树的数据仓库检索到的信息的装置。

25. 一种用于生成数据并从数据仓库访问数据的方法, 包括:

响应接收数据的情况，创建基于树的数据仓库的至少一个级别，
基于树的数据仓库的至少一个级别包括第一树、第二树以及至少第三
树，所述第一树中包括第一根和多个节点中的至少一个节点，所述第
二树中包括第二根和第一树的至少一个节点，所述第三树中包括第三
5 根和第一树的多个节点中的至少一个节点；

接收对可从基于树的数据仓库的至少一个级别访问的信息的信息
请求；以及

从基于树的数据仓库的至少一个级别中检索信息。

26. 根据权利要求 25 所述的方法，包括：

10 响应接收到要被添加到基于树的数据仓库中的数据，在基于树的
数据仓库中创建用于存储与接收到的数据关联的信息的新节点；

创建从新节点到包括新节点的第一部分的第一节点和到包括新
节点的第二部分的第二节点的链接；

将到新节点的链接添加到第一节点的第一指针列表中；以及

15 将到新节点的链接添加到第二节点的第二指针列表中。

27. 根据权利要求 26 所述的方法，其中，在基于树的数据仓库
中创建新节点包括将代表开始指示符的第一节点与代表数据集元素
的第二节点组合，以生成代表不完整的产物的第三节点。

28. 根据权利要求 27 所述的方法，其中，在基于树的数据仓库
20 中创建新节点包括将代表第一不完整的产物的第一节点与代表数据
集元素的第二节点组合，以生成代表第二不完整的产物的第三节点。

29. 根据权利要求 27 所述的方法，其中，在基于树的数据仓
库中创建新节点包括将第一不完整的产物与代表结束指示符的第四
节点组合，以生成代表最终产物的第五节点。

25 30. 根据权利要求 27 所述的方法，其中，在基于树的数据仓
库中创建新节点包括将第二不完整的产物与代表结束指示符的第四
节点组合，以生成代表最终产物的第五节点。

31. 根据权利要求 26 所述的方法，其中，创建新节点包括为
第一指针、第二指针、第三指针和第四指针分配空间。

32. 根据权利要求 26 所述的方法, 其中, 第三指针和第四指针是 Null 指针。

33. 一种从基于树的数据仓库中访问信息的方法, 该方法包括:

5 响应接收对数据仓库中的信息的请求, 该信息请求包括至少一个约束, 数据仓库包括基于树的数据仓库的一个级别, 基于树的数据仓库的至少一个级别包括第一树、第二树以及至少第三树, 所述第一树中包括第一根和多个节点中的至少一个节点, 所述第二树中包括第二根和第一树的至少一个节点, 所述第三树中包括第三根和第一树的多个节点中的至少一个节点,

10 检索与第三根关联的节点的列表, 该节点列表中的每一个节点都包括第一部分和第二部分, 其中, 第三根包括至少一个约束; 以及沿第一树中的每一个节点的树枝而行到第二树的至少一个节点。

34. 根据权利要求 33 所述的方法, 其中, 对信息的请求包括第一约束和第二约束, 并返回第二树的节点交集。

15 35. 包括用于生成互锁树数据库的计算机可执行的指令的计算机可读的介质, 包括用于执行下列操作的指令:

响应接收到要被添加到基于树的数据仓库中的数据, 在基于树的数据仓库中创建用于存储与接收到的数据关联的信息的新节点;

20 创建从新节点到包括新节点的第一部分的第一节点和到包括新节点的第二部分的第二节点的链接;

将到新节点的链接添加到第一节点的指针的第一列表中; 以及将到新节点的链接添加到第二节点的第二指针列表中。

36. 一种包括节点和所述节点之间的链接的结构, 所述节点具有多个数据字段, 所述多个数据字段中至少有两个数据字段包含指针, 所述至少两个指针中的一个 Case 指针, 所述至少两个指针的另一个是 Result 指针, 至少一个节点具有至少一个指向指针列表的附加指针, 在所述节点具有关联的 asCase 列表的情况下, 所述指向指针列表的附加指针中的一个 asCase 列表, 在所述节点具有关联的 asResult 列表的情况下, 另一个是 asResult 列表, 其中, 所述

节点包括根节点，其中，有至少一个主根节点和至少一个基本根节点，其中，所述节点可以包括其他根节点，所述节点进一步包括至少一个结束思考节点、至少一个子组件节点、以及至少一个最终产物节点，其中，所述 asResult 链接在根节点和任何其他节点之间的路径中在
5 所述根节点和所述任何其他节点之间提供指向，所述 asCase 链接在至少一个主根节点和至少一个最终产物节点之间提供指向，在所述至少一个主根节点和至少一个最终产物节点之间的路径上包含至少一个子组件节点，并且所述 asResult 链接在根节点或最终产物节点和子组件节点或最终产物节点之间在它们之间的所述路径上提供指向，
10 其中，所述基本节点还具有一个字段，该字段具有值。

37. 根据权利要求 36 所述的结构，所述结构由一组程序指令构成，该组程序指令当被激活时配置计算机系统，以产生所述结构。

38. 包含如权利要求 37 所述的程序指令集的计算机可读取的介质。

15 39. 包含符合权利要求 36 所述的结构的电子系统。

40. 根据权利要求 36 中所述的结构，其中，所述 asCase 和所述 asResult 列表存储在所述互锁树结构之外的单独的数据结构中，其中，所述单独的数据结构通过指针与所述互锁树结构中的相关的节点关联。

用于存储和访问互锁树数据 仓库中的数据的系统和方法

5

技术领域

本发明涉及计算领域，具体来说，涉及将数据存储和数据仓库中并访问其中的数据的领域。

背景技术

10

在开发软件时开发人员所作出的一个基本选择是选择适当的数据结构以便于组织和引用数据。有许多不同类型的数据结构，包括链接列表、堆栈、树、数组及其他。每一个数据结构都与某些优点和局限关联。

15

一个频繁使用的数据结构是树。树的一种常见形式由叫做节点的元素的有限集构成，从根到一个或多个内部节点链接在一起，其中每一个节点都可以链接到一个或多个节点，最终以许多叶节点结束。通常，与根比较近的节点是远离根的节点的父节点。远离根的节点叫做父节点的子节点。数据通常存储在节点中，并可以使用从根到节点到叶和从父节点到子节点的链接来引用。因此，分层或顺序关系可以是存储在树结构的节点中的数据的属性。分层关系也可以被理解为上下文关系，每一个节点都可在其父节点的上下文内被访问。

20

树数据结构的一个局限是，通常，一个树只能代表一个层次结构。例如，销售活动的根节点可以具有许多从根节点下垂的节点，每一个节点都代表特定销售员。每一个销售员节点都可以具有子节点，每一个销售员子节点都代表位于特定州的销售。因此，在销售员的上下文内可以轻松地访问此树，以查找州信息，即，此树可以用来有效地回答问题：“销售员 Bob 在哪个州进行销售？”如果不是按销售员访问州数据，而需求州的上下文内的销售员数据，（即，我们希望回答问题：“哪个销售员在得克萨斯州销售？”），则必须创建另一个树，代

25

表州的节点从根销售员活动下垂，代表销售员的子节点可以从该节点中下垂。创建另一个树的替代方法将是遍历整个树，以提取所希望的信息。

如果一个结构可以记录数据内的所有可能的上下文关系是有帮助的，从而实现用标准树数据结构无法实现的效率。

发明内容

生成和/或访问包括互连的树的林所构成的一个或多个级别的基于树的数据仓库。基于树的数据仓库的每一个级别包括从第一根节点下垂的第一树，并可以包括多个树枝。第一根可以代表诸如但不限于级别开始指示符之类的概念。第一树的每一个树枝都在叶节点中结束。每一个叶节点都可以代表最终产物，如下面比较全面地描述的。基于树的数据仓库的同一个级别的第二根链接到代表最终产物的第一树的每一个叶节点。因此，第二根基本上是第一树或第一树的子集的倒序的根，但第一树不重复。第二根可以代表诸如但不限于级别结束指示符之类的概念。最后，基于树的数据仓库包括多个树，其中，这些树中的每一个树的根节点都可以包括诸如数据集元素或数据集元素的表示法之类的数据。这种根节点这里被称为基本根节点。这些树中的每一个树的基本根节点都可以链接到非复制的第一树的一个或多个树枝中的一个或多个节点。基于树的数据仓库的非根节点只包含指向基于树的数据仓库中的其他节点的指针。包括基于树的数据仓库的每一个级别的树林中的树的根也由指示器组成，然而，根节点也可以包含代表信息的数据（即，包含是或代表诸如数据集元素之类的的数据或诸如级别开始或级别结束指示符之类的概念的数据）；基于树的数据仓库的所有其他节点只指向其他节点，不包含数据。在本发明的一个实施例中，数据是与字符关联的整数，像素表示法、诸如开始指示符、结束指示符、字段指示符的开始等等条件，尽管本发明不仅限于此。可以生成和访问上文所描述的基于树的数据仓库的多个级别；较低级别的最终产物变成下一级别的基本根节点。

生成和访问互锁树数据仓库。数据仓库包括构成了从第一根下垂

的一个 asCase 树的 asCase 树枝的多根树;这里叫做主根, asResult 树枝构成了从多重根下垂的多个 asResult 树。asResult 树的一个特殊实例包括链接到上文所描述的 asCase 树的一个或多个最终产物叶节点的根节点。因此, 此 asResult 树可以轻松地按照倒序访问在最终产物中结束的 asCase 树的树枝。此 asResult 树也可以用来定义下一级别的基本根节点。这些基本根节点可以代表由较低级别的最终产物集组成的下一级别的数据集元素。

互锁树数据仓库可以通过组合代表级别开始指示符的节点和代表数据集元素的节点以构成代表子组件的节点, 捕获有关在输入文件中遇到的数据集元素之间的关系的的信息。子组件节点可以同代表数据集元素的节点相组合, 以生成迭代子进程位于另一个子组件节点。将子组件节点与代表级别结束指示符的节点组合可以创建级别最终产物节点。将级别开始节点与数据集元素节点组合以创建子组件以及将子组件与数据集元素节点组合的过程本身可以迭代, 以生成一个级别中的多个 asCase 树枝。AsResult 树还可以通过指向 asCase 树中的一个或多个节点的 asResult 树的根, 链接或连接到 asCase 树中的节点。

一个级别的最终产物节点可以是代表数据集元素的基本根节点, 它们组合起来以生成子组件的下一级别。此过程可以重复任意次, 创建 asCase 树的任意个级别。另外, 还可以分解一个级别的基本根节点以生成较低级别的节点和根。一个级别的最终产物节点, 通过较低级别的 asResult 树的特殊实例, 即, 具有代表较低级别的结束指示符的根节点的较低级别的 asResult 树, 变成下一级别的基本根节点。如此, 具有代表较低级别的结束指示符的根节点的较低级别的 asResult 树是第二根, 较低级别的 asCase 树的反转。

在本发明的一个实施例中, 在创建节点时, 在每一个级别中基本上同时生成 asCase 和 asResult 链接。通过在处理输入时生成 asCase 链接来创建 AsCase 树枝。每一个级别上的 asCase 树的 asCase 树枝提供是如何通过将代表数据集元素的节点按顺序组合为

子组件等等到最终产物，创建级别每一个子组件和最终产物的直接记录。**asCase** 树的树枝还代表树中的节点的一个可能的分层关系。

asResult 链接的生成创建了一系列互锁树，每一个互锁树都从一个单独的根下垂。在一个级别中可能有多个此类型的根。这具有记录在输入中遇到的数据集元素之间的所有其他关系的结果。如前所述的信息是通过构建的而不是显式存储在树的节点中的互锁树的林的结构来捕获的，以便实际上作为输入接收的数据判断构建的互锁树的林的结构。**asResult** 树的林的结构确保了，可以在需要的任何其他上下文中访问如此存储的信息。因此，数据仓库是自组织的，从下面的描述中就可以看出。

附图说明

当与所附的图形阅读时，可以更好地了解前面的摘要，以及本发明的实施例的下面详细描述。为了说明本发明，在图形中显示了本发明的示范性结构；然而，本发明不仅限于所说明的特定方法。在图形中：

图 1 是其中可以实现本发明的各个方面的示范性计算环境；

图 2a 说明了根据本发明的一个实施例的用于从互锁树数据仓库生成数据并访问数据的示范性系统；

图 2b 说明了用于从互锁树形数据库生成信息并访问信息的示范性方法；

图 3a 显示了根据本发明的一个实施例的图 3a 的示范性互锁树数据仓库的比较详细的视图；

图 3b 显示了根据本发明的一个实施例的图 3a 的互锁树数据仓库的示范性节点的比较详细的视图；

图 3c 显示了根据本发明的一个方面的图 3a 的互锁树数据仓库的链接列表；

图 4 显示了根据本发明的一个实施例的存储在存储器中的图 2 的示范性数据集元素集；

图 5a-e 描述了图 2 的互锁树，以及根据本发明的一个实施例

生成互锁树时互锁树的节点的对应的内容;

图 6 是根据本发明的一个方面的生成互锁树的示范性过程的流程图;

图 7a 显示了根据本发明的一个实施例的另一个互锁树数据仓库和对应的节点;

图 7b 显示了根据本发明的一个方面的图 7a 的互锁树数据仓库的链接列表;

图 8 显示了根据本发明的实施例的其他互锁树数据仓库;

图 9a 显示了根据本发明的一个实施例的另一个互锁树数据仓库;

图 9b 显示了根据本发明的一个实施例的图 9a 的互锁树数据仓库的节点的示范性内容;

图 10 显示了根据本发明的一个实施例的另一个互锁树数据仓库; 以及

图 11 是根据本发明的一个实施例的从互锁树数据仓库访问数据的示范性过程的流程图。

具体实施方式

概述

如下所述的系统和方法创建了包括互连的树的林至少一个级别的数据仓库。数据仓库的每一个级别的互连的树的林捕获有关代表级别开始的节点和数据集元素(创建子组件节点)或子组件节点和数据集元素节点或子组件节点和代表迭代过程中的级别结束指示符的导致生成由被 asCase 树枝和多个 asResult 树链接的节点组成单个 asCase 树的节点的组合的信息。asCase 树枝的节点从第一根下垂。例如, 请参看图 3a, 节点 302、312、314 和 316 是从第一开始指示符根 302 下垂的示范性 asCase 树。AsResult 树包括下列树: 节点 306 和 312(一个 asResult 树), 节点 304 和 314(第二 asResult 树), 节点 308 和 316(第三 asResult 树)和节点 310 和 318(第四 asResult 树)。第四 asResult 树是 asResult 树的特殊实例, 因

为根（节点 310）代表了结束指示符。

给定级别的 asCase 树的每一个树枝以将代表级别开始指示符的节点和代表数据集元素的节点组合为子组件节点开始。子组件节点可以反复地同数据集元素节点相组合为另一个子组件节点。子组件可以同代表级别结束指示符的节点相组合，以创建最终产物节点。此过程可以重复，并可能导致形成从第一根下垂的多个 asCase 树枝。

例如，如果特定互锁树结构的不可分的基本组件是字母数字的，则子组件可以是不是单词的字母的组合，最终产物可以是单词。或者，子组件可以是包括部分库存品编号或订购编号的字母数字的组合，最终产物可以是完整的库存品或订购编号，这里只是举应用于本发明的输入的字母数字 universe 的两个可能的用途。

一个级别的最终产物可以是下一个级别的数据集元素。最终产物数据集元素可以用来生成子组件的下一级别，方式与较低级别的数据集元素用来创建较低级别的子组件和最终产物的方式相同。例如，在上文所描述的特定互锁树结构中，一个级别（单词）的最终产物可以是从中可以创建较高级别最终产物（句子）的数据集元素。此过程可以重复任意次，在数据仓库中创建 asCase 树的任意个级别。

为继续上文所描述的示例，较高的级别，使用单词作为级别数据集元素，可以包括句子。句子可以组合起来，以创建段落（较高的级别）等等。另外，还可以分解较高级别的数据集元素，以生成互锁树数据仓库的较低级别。在本发明的一个实施例中，从级别结束指示符启动的 asResult 树被用来定义下一级别的数据集元素。结束指示符是互锁树数据仓库的倒序的第二根，如本发明的一个实施例中的 asCase 树所定义的。

在创建节点时，在每一个级别中基本上同时生成 asCase 和 asResult 链接。asCase 链接代表到从中创建节点的两个节点的第一个节点的链接。可以理解，在处理输入时，通过生成 asCase 链接，可以创建 asCase 树的 asCase 树枝。每一个级别的 asCase 树枝提供是如何创建级别每一个子组件和最终产物的直接记录。因此，

asCase 树枝可以用于任何目的，知道子组件和最终产物是如何创建的是有用的。例如，如果到互锁树生成器的输入包括正确拼写的单词的 **universe**，则所生成的互锁树的所产生的 **asCase** 链接可以用作拼写检查程序，这里只是列出数据仓库的公用程序的许多可能的示例中的一个示例。

另外，**asCase** 树的树枝还代表 **asCase** 树中的节点的一个可能的分层关系。例如，如果互锁树生成器接收到的数据是“Tom 销售了 100 个 PA。Bill 销售了 40 个 NJ”，所生成的 **asCase** 树包括“销售员的上下文内的州信息”上下文或层次结构中的数据的视图。

asResult 链接代表到从中创建节点的两个节点的第二个节点的链接。**asResult** 链接的生成创建了一系列互锁树，其中，每一个 **asResult** 树都从包括数据集元素的根下垂。这具有记录数据仓库中的元素和 **asCase** 树之间的所有遇到的关系的结果。即，**asResult** 树捕获互锁树的节点的所有可能的上下文。**asResult** 树可以用于任何目的，知道节点之间的上下文或关系是有用的。例如，如果对互锁树数据仓库生成器的输入包括若干销售数据，包括销售员姓名、星期几、商品编号和州、所生成的互锁树数据仓库的所产生的 **asResult** 链接可以用来从同一个互锁树数据仓库中提取下列信息：“哪个销售员在特定的州销售？”“星期一销售了多少商品？”“销售员 Bob 在星期一和星期二销售了多少商品？”等等，无需创建数据仓库的多个副本。

可以使用存储在 **asResult** 树来分类子组件和最终产物。可以理解，如前所述的信息实际是通过构建的而不是显式存储在树的子组件和最终产物节点中的互锁树数据仓库的结构来存储的。由于只有互锁树数据仓库的根节点才可以包括数据，因此，可以沿 **asResult** 链接反向而行到根节点，以判断子组件或最终产物是否属于由根节点所代表的数据的类。根据如下所述的过程，进一步可以理解，此功能使数据仓库为自组织的。例如，如果对互锁树数据仓库生成器的输入是“CAT TAB”，则存储在所生成的互锁树数据仓库的结构中的信息可以用来判断两个最终产物“BOT-C-A-T-EOT”和“BOT-T-A-B-EOT”包

含元素“A”，或换句话说，包含“A”的子组件/最终产物的类包括“BOT-C-A-T-EOT”和“BOT-T-A-B-EOT”。此外，通过沿包含“A”的节点的 asCase 链接而行，可以沿着 asCase 树的树枝查找包含“A”的其他子组件和最终产物。

5 在本发明的一个实施例中，节点之间的链接是双向的。例如，代表字母“A”的根节点可以包括指向节点 A 的 asResultList 中的节点 BOT-C-A 的指针，而节点 BOT-C-A 可以包括指向节点 A 的指针作为其 asResult 指针，依次类推。

10 在本发明的另一个实施例中，节点之间的链接是单向的。例如，在本实施例中，节点 BOT-C-A 包括指向节点 BOT-C 的 asCase 指针和指向代表 A 根节点的 asResult 指针，但根节点 A 不包括指向其 asResultList 中的节点 BOT-C-A 的指针。那些精通本技术的人员将认识到，虽然仍可以判断涉及哪些节点属于类别 A 的信息，但是，这样做可能需要搜索所有节点。

15 示范性计算环境

20 图 1 是其中可以实现本发明的各个方面的示范性计算机系统 100 的方框图。计算机系统 100 可以是任何合适的系统，如但不限于大型机、小型计算机、IBM 兼容的个人计算机，Unix 工作站或网络计算机。那些精通相关技术的人将认识到，本发明的设备可以适用于任何计算机系统，包括但不限于多用户计算机系统或单用户计算机。如图 1 所示，计算机系统 100 包括连接到主存储器 104 的中央处理单元 (CPU) 102，辅助存储器接口 106，终端接口 108，以及网络接口 110。这些系统组件通过系统总线 160 连接起来。辅助存储器接口 106 用来连接存储设备，如但不限于 DASD 设备 190，将
25 存储在如但不限于磁盘 195 之类的磁盘上的数据仓库到计算机系统 100。

 包含了计算机系统 100 的整个虚拟存储器的主存储器 104，包括了操作系统 122 和应用程序 124，还可以包括互锁树数据仓库 126。互锁树数据仓库 126 可以用来提供这样的数据仓库：可以以多

个上下文模式对该数据仓库快速进行搜索，以查找数据，而不需要数据的重复。计算机系统 100 可以使用已知的虚拟寻址机制，这种机制允许计算机系统 100 的程序这样操作：似乎它们可以访问大型单个存储实体而不是访问诸如主存储器 104 和 DASD 设备 190 之类的多个较小的存储实体。因此，尽管显示的操作系统 122、应用程序 124，以及互锁树数据仓库 126 驻留在主存储器 104 上，那些精通相关技术的人将认识到，这些元件不一定都完全同时位于主存储器 104 中。

尽管显示的计算机系统 100 只包括单个 CPU 和系统总线，但是，那些精通相关技术的人将理解，可以使用包括多个 CPU 和/或总线的系统来实现本发明。可以使用终端接口 108 来将一个或多个终端连接到计算机系统 100。引用的终端可以是哑终端，或完全可编程的工作站，可以被用来使系统管理员和用户与计算机系统 100 进行通信。

可以使用网络接口 110 来将其他计算机系统和/或工作站连接到计算机系统 100。网络接口 110 所连接的网络可以是局域网 (LAN)、广域网 (WAN)，内部网、extranet 或因特网，或任何其他合适的网络。操作系统 122 可以是诸如 OS/2、WINDOWS、AIX、UNIX、LINUX 之类的操作系统或任何其他合适的操作系统。

应用程序 124 可以是访问存储在互锁树数据仓库 126 中的数据的任何类型的应用程序。如此，应用程序可以包括数据分析应用程序、数据仓储应用程序、入侵检测系统，这里仅举几个例子，虽然本发明不仅限于此。

互锁树数据仓库 126 提供能使用户访问相同数据仓库以获取与任何上下文关联的信息的数据仓库结构。这里所使用的术语“数据”可以包括任何类型的计算机存储的信息，如但不限于数字、文本、图形、公式、表、音频、视频、多媒体或其任何组合。互锁树数据仓库 126 可以作为应用程序 124 的一部分、作为操作系统 122 的一部分或作为可以用于为各式各样的应用程序提供数据仓库的单独的数据仓库

产品。

尽管是在完全运转的计算机系统的上下文中描述本发明的，但是，那些精通本技术的普通人员将认识到，本发明能够作为各种各样的形式的程序产品进行分发，本发明同样适用，不管用于进行分发的
5 信号携带介质的特定类型是什么。携带这样的信号的介质的示例包括但不仅限于软盘、硬盘驱动器、CD-ROM、电的、光学、无线的数字和模拟通信链路或其他合适的介质。

用于生成和访问互锁树数据仓库的系统和方法

图 2a 说明了根据本发明的一个实施例的用于从构成了数据仓库
10 的互锁树林生成数据并访问数据的示范性系统 200。一个实施例中的用于生成互锁树数据仓库的子系统 250 包括互锁树生成器 202、一组数据集元素 206，用来生成示范性互锁树数据仓库 208 的输入数据 204。或者，可以从输入数据 204 派生数据集元素集 206。

用于从互锁树数据仓库 208 访问信息的子系统 251 可以包括
15 互锁树数据仓库 208，如上所述，和/或 用于接收数据请求 212、处理数据请求 212 并返回请求的信息的互锁树数据仓库存取器 210。

图 2b 说明了用于从互锁树数据库生成信息并访问信息的示范性方法。在步骤 260 中，如下面比较全面地描述的，生成互锁树数据仓库。在步骤 262 中，接收需要互锁树数据仓库中的信息的请求。
20 在步骤 264 中，从互锁树数据仓库中检索信息。

生成互锁树数据仓库

例如，假设输入数据 204 包括代表一个单词（例如，“CAT”）的字母数字字符的流。在此情况下，数据集元素 206 可以是字母表中的字母集，并可以包括一个或多个字符，以代表分隔符或开始单词
25 /单词结束的概念。分隔符可以包括字母数字字符，如但不限于空白（“ ”）、逗号（“，”），和句点（“.”）。

互锁树数据仓库 208 包括许多根、许多非根节点和非根节点之间的或根和非根节点之间的许多链接或连接。互锁树数据仓库 208 的每一个根和非根节点都包括一对指针（情况指针(case pointer) 和结

果指针(result pointer))和一对列表指针(指向 asCaseList 的指针和指向 asResultList 的指针)。此外,根还可以包括代表值或对值的引用的数据。

图 3a 是示范性互锁树数据仓库 208 的比较详细的视图。某些节点,特别是,在本示例中,根节点 302 (BOT) 和 310 (EOT),代表诸如开始指示符或结束指示符之类的概念,根节点 304(A)、306(C)、308(T) 代表数据集元素,而其他节点,特别是,节点 312 (BOT-C)、314 (BOT-C-A)、316 (BOT-C-A-T) 和 318 (BOT-C-A-T-EOT) 代表了代表开始指示符的节点和代表数据集元素的节点顺序合成为代表子组件的节点,该子组件与数据集元素组合成为另一个子组件,依次类推,直到子组件与代表结束指示符的节点组合,从而创建代表最终产物的节点。在此情况下,捕获来自一连串字母(后面是分隔符(即,一连串字母“CAT”,后面是分隔符“ ”或空白字符))的单词的顺序合成。输入中的分隔符可以用来区别最终产物。例如,值得注意的是,分隔单词的字符可以指出一个单词的末尾和另一个单词的开始。例如,在字符串“CATS ARE”中,“CATS”和“ARE”之间的空白字符既表示单词“CATS”的末尾,又表示单词“ARE”的开始。因此,诸如输入中的空白字符之类的分隔符可以被创建的节点中的诸如“BOT”之类的开始指示符替代,或被诸如“EOT”之类的结束指示符替代,如下面比较全面地描述的。

诸如根节点 304、306 和 308 之类的节点这里被称为基本节点,因为这些节点代表数据集元素,并包括用来构成可分的单元(子组件和最终产物)的不可分的单元。诸如 312、314 和 316 之类的节点这里被称为子组件或子组件节点,因为这些节点代表诸如开始指示符之类的概念指示符和代表数据集元素的节点的组合,或子组件和代表数据集元素的不包括最终产物的节点的组合或子组件和代表结束指示符的包括最终产物的节点的组合。诸如节点 318 之类的节点代表最终产物。在所引用的示例中,数据集元素是字母,子组件代表了不包括单词的字母的组合,最终产物是单词。值得注意的是,根节

点集包括在本示例中表示单词开始的“BOT”和表示单词末尾的“EOT”。可以理解，“BOT”和“EOT”代表首尾指示符，但本发明不仅限于此。也可以使用其他这样的指示符，如同不存在一个或两个这样的指示符。在本发明的一个实施例中，由于从节点到代表 EOT 概念的根节点而将最终产物与子组件区分开。

可以理解，尽管在给定示例中，输入的 universe 是可用派生单词的字母数字字符集，但是，本发明没有这样的限制。例如，输入的 universe 可以是文本，如字母（可以用来派生单词）或单词（可以用来派生词组或句子），或者，也可以是用来派生基因组的氨基酸，进程中使用的有限资源，概念、像素集、图像、声音、数字、模拟测量或值或由基本单元构成的可以被数字化和按顺序组合以生成最终产物的任何其他合适的 universe。通常，根据本发明的一个实施例，基本单元以优化的序列组合。

除了上文所描述的节点之外，互锁树数据仓库 208 还可以包括节点之间的许多连接或链接，如链接 320、322、324 和 326，以及链接 328、330、332 和 334。本发明的一个实施例中的链接 320、322、324 和 326 以及链接 328、330、332 和 334 是双向的，即，根节点 (BOT) 和节点 318 (BOT-C-A-T-EOT) 之间的通道通过链接 320、322、324 和 326 遍历，或者，也可以通过链接 326、324、322 和 320 来遍历。链接 320、322、324 和 326（用实线表示）这里被称为 asCase 链接。链接 328、330、332 和 334（用虚线表示）这里被称为 asResult 链接。同样，在本发明的一个实施例中，链接 328、330、332 和 334 是双向的，因为节点 C 306 中的指针指向节点 BOT-C 312，节点 BOT-C 312 中的指针指向节点 C 306，节点 A 304 中的指针指向节点 BOT-C-A 314，节点 BOT-C-A 314 中的指针指向节点 A 304 等等。

图 3b 显示了互锁树数据仓库 208 的示范性节点中包括的信息。示范性节点 340 可以代表子组件或最终产物。示范性节点 340 可以包括指向子组件或最终产物 340 的第一部分的指针（指向 case

342 的指针，这里也被称为“asCase”），指向子组件或最终产物 340 第二部分的指针（指向结果 344 的指针，这里也被称为“asResult”），指向 asCaseList 346 的指针，其节点 340 是第一部分和指向 AsResultList 348 的指针的子组件或最终产物的链接列表，其节点 5 340 是第二部分的组件或最终产物的链接列表。

示范性节点 341 代表根节点。示范性节点 341 包括 Null 指针（指向 case 342 的指针，这里也被称为“asCase”），第二 Null 指针（指向结果 344 的指针，这里也被称为“asResult”），指向 asCaseList 346 的指针，其根节点 341 是第一部分和指向 10 AsResultList 348 的指针的子组件或最终产物的链接列表，其根节点 341 是第二部分和值 349 的组件或最终产物的链接列表。值 349 可以包含实际值，代表条件或状态，可以包含指针或对值的引用等等。因此，在本发明的一个实施例中，代表开始指示符概念或条件的根节点将具有空值 asResultList，因为开始指示符永远不会是子组件的第二部分，代表数据仓库元素的根节点将具有空值 asCaseList，因为数 15 据集元素永远不会是子组件的第一部分，代表结束指示符概念或条件的根节点将具有空值 asCaseList，因为结束指示符永远不会是子组件的第一部分。最后由较低级别的最终产物组成的根节点将具有空值 asCaseList，因为最终产物充当下一级别的数据集元素。

20 图 3c 显示了互锁树数据仓库 208 的 asResult 链接列表。链接 350 是通过设置节点 C 306 到节点 BOT-C 302 的 asResultList 中的指针而建立的，链接 352 通过设置节点 A 304 到节点 BOT-C-A 314 的 asResultList 中的指针而建立的，链接 354 是通过设置节点 T 308 到节点 BOT-C-A-T 318 的 asResultList 中的指针而建立的链接 356 是通过节点 EOT 310 到节点 25 BOT-C-A-T-EOT 318 的 asResultList 中的指针而建立的。

图 4 描述了存储器 104 中的示范性数据集元素 206 BOT、A-Z 和 EOT 的示范性存储。可以看出，在本示例中，BOT 存储在位置 0，A 存储在位置 5，等等，EOT 存储在 135。可以理解，数

数据集元素的放置只是示范性的，数据集元素的任何合适的放置都是可以的。图 5a-e 描述了互锁树数据仓库 208 和互锁树数据仓库 208 的节点的对应的内容，互锁树数据仓库 208 是在本发明的示范性实施例中生成的。图 6 是根据本发明的一个实施例的用于生成互锁树数据仓库 208 的示范性过程 600 的流程图。

请同时参看图 4、5 和 6，在步骤 602 中，初始化互锁树数据仓库。在本发明的一个实施例中，初始化包括将“当前指针”设置到将要创建的互锁树数据仓库的根节点。在本发明的另一个实施例中，初始化包括将“当前指针”设置到现有的互锁树数据仓库的根。

此外，数据集元素可以加载到存储器中，并进行初始化。在本发明的一个实施例中，用下列值初始化根节点（例如，图 5a 的根节点 BOT 302，A 535a，... EOT 559a）：情况指针 = 空值，结果指针 = 空值，指向 asCaseList 的指针 = 空值，asCaseList = 空值，指向 asResultList 的指针 = 空值，asresultlist = 空值，以及数据集元素或概念/条件指示符或表示法的值。

此时，互锁树数据仓库，如，根据本发明的一个实施例的互锁树数据仓库 500a，可以包括单个节点 302 (BOT)，在此情况下，表示一个单词的开始。方框图 502a 的节点 302 包括一对指针（被初始化为空值的情况指针 504a 和结果指针 506a）和一对列表指针（被初始化为空值的指向 asCaseList 的指针和指向 asResultList 的指针）和值（被初始化为某个值的值 511a，这里描述为 BOT）。为便于理解，在图 5 中，方框图 502a，图 502b-e 中的单元 508a 和类似的单元，互锁树数据仓库实际上代表了指向关联的 asCaseList 的指针的所有图，显示了关联的 asCaseList 的当前内容。同样，互锁树数据仓库实际上代表了指向关联的 asResultList 的指针的图 502b-e 中的单元 510a 和类似的单元，显示了关联的 asResultList 的当前内容。

AsCaseList（例如，asCaseList 508a）和 AsResultList（例如，asResultList 510a）可以作为链接列表来实现。在另一个实施例中，

asCaseList (例如, **asCaseList 508a**) 和 **AsResultList** (例如, **asResultList 510a**) 作为大小可配置的连续的存储器位置的块来分配, 如但不限于数组, 指向 **asCaseList** 的指针被设置为 **asCaseList** 存储块的开始位置, 指向 **asResultList** 的指针被设置为 **asResultList** 存储块的开始位置。

在步骤 604 中, 接收输入。在本发明的一个实施例中, “当前指针”的值被设置为“前一指针”, “当前指针”被设置为输入。在给定的示例中, 接收到的输入为“C”。在步骤 604 中, 对输入进行验证。在给定的示例中, 这涉及检查“C”是否为有效的数据集元素。“C”的确是有效的元素, 位于存储器 104 中的位置 15 中。

在步骤 606 中, 如果节点已经不存在, 则创建互锁树数据仓库中的节点, 将其初始化, 并存储在存储器中的某个位置。在示例中, 创建互锁树数据仓库 208 中的节点 312, 代表 BOT-C, 节点 BOT-C 312 的情况指针、结果指针、指向 **asCaseList** 的指针、**asCaseList**、指向 **asResultList** 的指针, 以及 **asResultList**, 被初始化为空值, BOT-C 存储在存储器 104 的位置 140 处。

在步骤 608 中, 根据本发明的一个实施例, 创建在步骤 606 中创建的节点的链接。通过将新节点的情况指针设置为前一指针的值, 并将新节点的结果指针设置为当前指针的值, 来定义新节点。图 5b 的互锁树数据仓库 500b 显示了创建链接之后的互锁树数据仓库 208。方框图 502b 显示了创建链接之后的节点 BOT 302、C 306 和 BOT - C 312 的内容。通过将节点 BOT 302 与节点 C 306 顺序组合来创建子组件 BOT-C 312。因此, 设置了情况指针和结果指针的下列值: 节点 BOT-C 312 的情况指针 520b 被设置为 0 (存储器 104 中的节点 BOT 302 的位置), 节点 BOT-C 312 的结果指针 522b 被设置为 15 (存储器 104 中的基本节点 C 306 的位置)。

在本发明的一个实施例中, 除了创建从新节点到从中派生的新节点的节点的链接, 通过将指向新节点的位置的指针添加到从中派生了新节点的节点的链接列表, **asCaseList** 和 **asResultList**, 创建

asCaseList 和 **asResultList** 链接。指针可以添加到列表的末尾，添加到列表的开始，也可以插入列表内的某处。另外，也可以维护许多列表。例如，除了其中按照最频繁地被访问的顺序维护了指针的有序列表之外，节点的 **asCaseList** 可以包括其中指针被添加到链接列表的末尾的顺序列表。可以理解，虽然给定的示例引用了一个有序列表和一个顺序列表，但是，本发明不仅限于此，列表的任何组合和任何数量都在本发明的范围内。有序列表可以按照最近更新、最后一次的访问、或更新或访问的频率，或按照任何其他合适的排序规则来进行排序。

10 建立到新节点的链接：将指向新节点的指针添加到前一指针的 **asCaseList**，并添加到当前指针的 **asResultList**。在本示例中，通过将节点 **BOT-C 312** 的情况指针 **520b** 设置为节点 **BOT 302** 的位置（位置 0）来生成双向链接 **320**（方框图 **503b** 的链接 **320a**），通过将指向节点 **BOT-C 312** 的位置（位置 140）的指针添加到
15 **asCaseList 508b** 来更新节点 **BOT 302** 的 **asCaseList 508b**（链接 **320b**）。设置了情况指针 **520a**，因为节点 **BOT 302** 是节点 **BOT-C 312** 的定义节点之一。更新了 **AsCaseList 508b**，因为在合成从中创建节点 **BOT-C 312** 的两个节点中的第一个节点（节点 **BOT-C 312**）时使用节点 **BOT 302**。**AsCaseList 508b** 目前包含空集（即，
20 **asCaseList 508b** 是空的）。因为节点 **BOT-C 312** 位于存储器 **104** 中的位置 **140**，因此，**asCaseList 508b** 从空值更新为 **140**。假设 **asCaseList 508b** 包括非零集，则节点 **BOT-C 312** 位置 **140** 将以上文讨论的任何一种方式添加到 **asCaseList 508b** 中。

同样，通过将节点 **BOT-C 312** 的结果指针 **522b** 设置为节点
25 **C** 的位置（位置 **15**）来生成双向链接 **328**（方框图 **503b** 的链接 **328a**），通过将指向节点 **BOT-C 312** 的位置的指针添加到 **asResultList 518b** 来更新基本节点 **BOT 306** 的 **asResultList 518b**（链接 **328b**）。设置了结果指针 **522b**，因为节点 **C 306** 是节点 **BOT-C 312** 的定义节点之一。更新了 **AsResultList 518b**，因为节点

C 306 包括从中创建节点 BOT-C 312 的两个节点的第二个节点,(因此,链接 328b 叫做 asResult 链接)。AsResultList 518b 目前包含空集,(即,asResultList 518b 是空的)。因为节点 BOT-C 312 位于存储器 104 中的位置 140,因此,asResultList 518b 从空值更新
5 为 140。假设 asResultList 518b 包括非零集,则节点 BOT-C 312 位置 140 将以上文讨论的任何一种方式添加到 asResultList 518b 中。

此时,已经创建了图 5b 中描述的数据仓库,互锁树数据仓库 500b。在图 5b 中的方框图 503b 中比较详细地显示了相同结构。值得注意的是,链接 320b 代表了指向节点 BOT-C 312 的位置的指针,并是节点 BOT 302 的 asCaseList 508b 中的第一个元素,链接
10 328b 代表指向节点 BOT-C 312 的位置的指针,并是节点 C 306 的 asResultList 518b 中的第一个元素。链接 320a 代表了从节点 BOT-C 312 到其第一部分(节点 BOT 302)的指针,而链接 328a 代表了从节点 BOT-C 312 到其第二部分(节点 C 306)的指针。

15 在步骤 610 中,判断是否有更多输入。在此情况下,有更多输入,因此,处理返回到步骤 604。在步骤 604 中,接收输入。在给定的示例中,接收到的输入为“A”。在步骤 604 中,对输入进行验证。在给定的示例中,这涉及检查“A”是否为有效的数据集元素。“A”的确是有效的元素,位于存储器 104 中的位置 5 中。

20 在步骤 606 中,如果节点已经不存在,则创建互锁树数据仓库中的节点,将其初始化,并存储在存储器中的某个位置。在示例中,创建互锁树数据仓库 208 中的节点 314,代表 BOT-C-A、节点 BOT-C-A 314 的情况指针、结果指针、指向 asCaseList 的指针、asCaseList、指向 asResultList 的指针,以及 asResultList,被初始
25 化为空值,节点 BOT-C-A 314 存储在存储器 104 的位置 145 处。

在步骤 608 中,根据本发明的一个实施例,创建在步骤 606 中创建的节点的链接。图 5c 显示了创建链接之后的互锁树数据仓库 500c。方框图 502c 中显示了节点 BOT 302、C 306、A 304、BOT-C 312 和 BOT-C-A 314 的内容。通过将节点 BOT-C 312 与节点 A

304 顺序组合来创建子组件 BOT-C-A 314。因此，设置了情况指针和结果指针的下列值：节点 BOT-C-A 314 的情况指针 528c 被设置为 140(链接 322a)，这是存储器 104 中的基本节点 BOT-C 312 的位置，节点 BOT-C-A 314 的结果指针 530c 被设置为 5(链接 330a)，这是存储器 104 中的基本节点 A 304 的位置。

通过将情况指针 528c 设置为 140(链接 322a)并通过将指向存储器 104 中的节点 BOT-C-A 314 的位置的指针添加到节点 BOT-C 312 的 asCaseList 524c(链接 322b)，生成双向链接 322。更新了 AsCaseList 524c，因为节点 BOT-C 312 包括从中创建节点 BOT-C-A 314 的两个节点中的第一个节点。在创建链接 322b 之前，节点 BOT-C 312 的 asCaseList 524c 包含了空集(即，asCaseList 524c 是空的)。因为节点 BOT-C-A 314 位于存储器 104 中的位置 145，因此，asCaseList 524c 从空值更新为 145。假设 asCaseList 524c 包括非零集，则节点 BOT-C-A 314 位置 145 将以上文讨论的任何一种方式添加到 asCaseList 524c 中。

同样，通过将节点 BOT-C-A 314 的结果指针 530c 设置为 5，并通过将指向节点 BOT-C-A 314 的位置的指针添加到节点 A 304 的 asResultList 542c 而更新基本节点 A 304 的 asResultList 542c，生成双向链接 330。更新了 AsResultList 542c，因为节点 A 304 包括从中创建节点 BOT-C-A 314 的两个节点中的第二个节点。在创建链接 330b 之前，AsResultList 542c 包含了空集，(即，AsResultList 542c 是空的)。因为节点 BOT-C-A 314 位于存储器 104 中的位置 145，因此，asResultList 542c 从空值更新为 145。假设 AsResultList 542c 包括非零集，则节点 BOT-C-A 314 位置 145 将以上文讨论的任何一种方式添加到 AsResultList 542c 中。

此时，已经创建了图 5c 中描述的数据仓库，互锁树数据仓库 500c。在图 5c 中的方框图 503c 中比较详细地显示了相同结构。值得注意的是，链接 322b 代表了指向节点 BOT-C-A 314 的位置的指针，并是节点 BOT-C 312 的 asCaseList 524c 中的第一个元素，

链接 330b 代表指向节点 BOT-C-A 314 的位置的指针，并且 145 是节点 A 304 的 AsResultList 542c 中的第一个元素。链接 322a 代表了从节点 BOT-C-A 314 到其第一部分(节点 BOT-C 312)的指针，而链接 330a 代表了从节点 BOT-C-A 314 到其第二部分(节点 A 304)的指针。

在步骤 610 中，判断是否有更多输入。在此情况下，有更多输入，因此，处理返回到步骤 604。在步骤 604 中，接收输入。在给定的示例中，接收到的输入为“T”。在步骤 604 中，对输入进行验证。在给定的示例中，这涉及检查“T”是否为有效的数据集元素。“T”的确是有效的数据集元素，位于存储器 104 中的位置 100 中。

在步骤 606 中，如果节点已经不存在，则创建互锁树数据仓库中的节点，将其初始化，并存储在存储器中的某个位置。在示例中，创建互锁树数据仓库 208 中的节点 316，代表 BOT-C-A-T 316，情况指针、结果指针、指向 asCaseList 的指针、asCaseList、指向 asResultList 的指针，以及 asResultList，被初始化为空值，节点 BOT-C-A-T 316 存储在存储器 104 的位置 150 处。

在步骤 608 中，创建在步骤 606 中创建的节点的链接。图 5d 显示了创建链接之后的互锁树数据仓库 500d。方框图 502d 中显示了节点 BOT 302、C 306、A 304、T 308、BOT-C 312、BOT-C-A 314 和 BOT-C-A-T 316 的内容。通过将节点 BOT-C-A 314 与节点 T 308 顺序组合来创建子组件 BOT-C-A-T 316。因此，为节点 BOT-C-A-T 316 设置了情况指针和结果指针的下列值：情况指针 544d 被设置为 145(存储器 104 中的节点 BOT-C-A 314 的位置)，结果指针 546d 被设置为 100(存储器 104 中的基本节点 T 308 的位置)。

通过将情况指针 544d 设置为 140 并将指向存储器 104 中的节点 BOT-C-A 314 的位置(位置 150)的指针添加到节点 BOT-C-A 314 的 asCaseList 532d，生成双向链接 324。更新了 AsCaseList 532d，因为节点 BOT-C-A 314 包括从中创建节点 BOT-C-A-T 316

的两个节点中的第一个节点。在创建链接 324 之前, 节点 BOT-C-A 314 的 asCaseList 532d 包含空集。因为节点 BOT-C-A-T 位于存储器 104 中的位置 150, 因此, asCaseList 532d 从空值更新为 150。假设节点 BOT-C-A 314 的 asCaseList 532d 包含数据, 则 150 将
5 以上文概述的任何一种方式添加到列表中。

同样, 通过将结果指针 546d 设置为 100 并通过将指向节点 BOT-C-A-T 316 的位置的指针添加到 asResultList 558d 来更新 asResultList, 生成双向链接 332。更新了 AsResultList 558d, 因为
10 节点 T 308 包括从中创建节点 BOT-C-A-T 316 的两个节点中的第二个节点。在创建链接 332 之前, 基本节点 T 308 的 asResultList 558d 包含空集, 因此, 空集被替换为 150 (存储器 104 中的节点 BOT-C-A-T 316 的位置)。假设 asResultList 558d 包含数据, 则 150 将以上文概述的任何一种方式添加到列表中。

此时, 已经创建了图 5d 中描述的数据仓库, 互锁树数据仓库
15 500d。那些精通本技术的人员将理解, 可以显示互锁树数据仓库 500d 的比较详细的表示法, 类似于图 5c 中的互锁树数据仓库 500c 的方框图 503c 所描述的。

在步骤 610 中, 判断是否有更多输入。在此情况下, 有更多输入, 因此, 处理返回到步骤 604。在步骤 604 中, 接收输入。在给
20 定的示例中, 接收到的输入为“”或空白字符。在步骤 604 中, 对输入进行验证。在给定的示例中, 这涉及检查本空白字符是否为有效的数据集元素。空白字符的确是有效元素在此情况下, 是表示单词“CAT”的末尾的分隔符。如此, 在本发明的一个实施例中, 位于位置 135 处的节点 EOT 310 被添加到子组件 BOT-C-A-T 316 以创建最
25 终产物或 monad, 在此情况下是一个单词。

在步骤 606 中, 如果节点已经不存在, 则创建互锁树数据仓库中的节点, 将其初始化, 并存储在存储器中的某个位置。在示例中, 创建互锁树数据仓库 208 中的节点 318, 代表节点 BOT-C-A-T-EOT 318, 节点 BOT-C-A-T-EOT 318 的情况指针、结

果指针、指向 **asCaseList** 的指针、**asCaseList**、指向 **asResultList** 的指针和 **asResultList** 被初始化为空值，节点 **BOT-C-A-T-EOT 318** 存储在存储器 104 的位置 150 处。

在步骤 608 中，创建在步骤 606 中创建的节点的链接。图 5e 显示了创建链接之后的互锁树数据仓库 500e。方框图 502e 显示了创建链接之后的节点 **BOT 302**、**C 306**、**A 304**、**T 308**、**EOT 310**、**BOT-C 312**、**BOT-C-A 314**、**BOT-C-A-T 316** 和 **BOT-C-A-T-EOT 318** 的内容。通过将节点 **BOT-C-A-T 316** 与节点 **EOT 310** 顺序组合来创建最终产物 318 (**BOT-C-A-T-EOT**)。因此，设置了节点 **BOT-C-A-T-EOT 318** 的情况指针和结果指针的下列值：最终产物 **BOT-C-A-T-EOT 318** 的情况指针 568e 被设置为 150（存储器 104 中的节点 **BOT-C-A-T 316** 的位置），最终产物 **BOT-C-A-T-EOT 318** 的结果指针 570e 被设置为 135（存储器 104 中的元素节点 **EOT 135** 的位置）。

通过将最终产物 **BOT-C-A-T-EOT 318** 的情况指针 568e 设置为 150 并将指向存储器中的节点 **BOT-C-A-T 316** 的位置的指针添加到节点 **BOT-C-A-T 316** 的 **asCaseList 548e**，生成双向链接 326。更新了 **AsCaseList 548e**，因为节点 **BOT-C-A 314** 包括从中创建节点 **BOT-C-A-T 316** 的两个节点中的第一个节点。在创建链接 334 之前，节点 **BOT-C-A-T 316** 的 **asCaseList 548e** 包含了空集（即，**asCaseList 548e** 是空的）。因为节点 **BOT-C-A-T 316** 位于存储器 104 中的位置 155，因此，节点 **BOT-C-A-T 316** 从空值更新为 155。假设 **asCaseList 548e** 包括非零集，则节点 **BOT-C-A-T** 位置 155 将以上文讨论的任何一种方式添加到 **asCaseList 548e** 中。

同样，通过将最终产物 **BOT-C-A-T-EOT 318** 的结果指针 570e 设置为 135 以及通过将指向节点 **BOT-C-A-T-EOT 318** 的位置的指针添加到节点 **EOT 310** 的 **asResultList 566e** 来更新节点 **EOT 310** 的 **asResultList 566e**，生成双向链接 334。更新了 **AsResultList 566e**，因为节点 **EOT 310** 包括从中创建节点

BOT-C-A-T-EOT 318 的两个节点的第二个节点,(因此,链接 334 叫做 asResult 链接)。在创建链接 334 之前, AsResultList 566e 包含了空集, (即, AsResultList 566e 是空的)。因为节点 BOT-C-A-T-EOT 318 位于存储器 104 中的位置 155, 因此,
5 asResultList 566e 从空值更新为 155。假设 AsResultList 566e 包括非零集, 则节点 BOT-C-A-T-EOT 318 位置 155 将以上文讨论的任何一种方式添加到 AsResultList 566e 中。

此时, 已经创建了图 5e 中描述的数据仓库, 互锁树数据仓库 500e。那些精通本技术的人员将理解, 可以显示互锁树数据仓库 500e
10 的比较详细的表示法, 类似于图 5c 中的互锁树数据仓库 500c 的方框图 503c 所描述的。

在步骤 610 中, 判断是否有更多输入。在此情况下, 没有更多输入, 因此, 处理在步骤 612 中结束。

现在假设输入 204 包含“CAT TAB”而不是“CAT”。按照上面的
15 过程执行。在处理输入直到“CAT”时, 创建图 5e 的互锁树数据仓库 500e。然而, 在步骤 610 中, 有更多输入, 因此, 处理继续, 最终产生图 7a 的互锁树数据仓库 700a。方框图 702a 中显示了对应的节点 BOT 302、C 306、A 304、T 308、B 718、EOT 310、BOT-C 312、BOT-C-A 314、BOT-C-A-T- 316、BOT-C-A-T-EOT 318、BOT-T 703、
20 BOT-T-A 705、BOT-T-A-B 707 和 BOT-T-A-B-EOT 709。值得注意的是, 节点 BOT-T 703、BOT-T-A 705、BOT-T-A-B 707 和 BOT-T-A-B-EOT 709 都已经添加到互锁树数据仓库 500e 中, 以创建互锁树数据仓库 700a。

在此过程中, 创建了 asCase 链接 701、704、706 和 708, 创
25 建了 asResult 链接 710、712、714 和 716。节点 BOT-T 703 的 AsCase 指针 720f 被设置为 0 (节点 BOT 302 的位置)。节点 BOT-T 703 的 AsResult 指针 722f 被设置为 100 (节点 T 308 的位置)。节点 BOT-T-A 705 的 AsCase 指针 728f 被设置为 170 (节点 BOT-T 703 的位置)。节点 BOT-T-A 705 的 AsResult 指针

730f 被设置为 5 (节点 A 304 的位置)。

通过将 170 (BOT-T 703 的位置) 和节点 BOT 302 的 asCaseList 508f 相加来创建 AsCase 链接 701, 以便 asCaseList 508f 包括 140 (BOT-C 312 的位置) 和 170 (BOT-T 703 的位置) 5 两者。通过将 175 (BOT-T-A 的位置) 和节点 BOT-T 703 的 asCaseList 724f 相加来创建 AsCase 链接 704。通过将 180 (BOT-T-A-B 的位置) 和节点 BOT-T-A 705 的 asCaseList 732f 相加来创建 AsCase 链接 706。

通过将 170 (BOT-T 703 的位置) 和节点 T 308 的 10 asResultList 558f 相加来创建 AsResult 链接 710, 以便 asResultList 558f 包括 150 (节点 BOT-C-A-T 的位置) 和 170 (BOT-T 703 的位置) 两者。通过将 175 (BOT-T-A 的位置) 和节点 A 304 的 asResultList 542f 相加来创建 AsResult 链接 712, 以便 asResultList 542f 包括 145 (节点 BOT-C-A 314 的位置) 和 175 15 (BOT-T-A 的位置) 两者。通过将 180 (节点 BOT-T-A-B 707 的位置) 和节点 B 718 的 asResultList 742f 相加来创建 AsResult 链接 714。因为节点 B 718 的 asResultList 742f 以前是空值, 因此, 节点 B 718 的 asResultList 742f 只包含 180 (节点 BOT-T-A-B 707 的位置)。通过将 185 (BOT-T-A-B-EOT 709 的位置) 和节点 EOT 20 310 的 asResultList 566f 相加来创建 AsResult 链接 716, 以便 asResultList 566f 包括 155 (节点 BOT-C-A-T-EOT 318 的位置) 和 185 (BOT-T-A-B-EOT 185 的位置)。

现在假设输入 204 包含“CATS CATHODE”而不是“CAT”。按照上面的过程执行。在处理输入“CAT”时, 创建图 5d 的互锁树数据 25 仓库。在步骤 610 中, 发现有更多输入, 因此, 处理继续。在处理输入“CATS”之后, 生成了图 8 的互锁树数据仓库 800a。发现有更多输入。在处理“CATHODE”时, 没有创建 BOT-C、BOT-C-A 和 BOT-C-A 的新节点, 因为它们已经存在。处理额外的输入“S CATHODE”, 产生图 8 的互锁树数据仓库 800b。对那些精通本技

术的人员显而易见的，所产生的树是自组织的，以便所产生的互锁树数据仓库的结构取决于接收到的输入。

现在假设输入 204 包含“CATS ARE FURRY”而不是“CAT”。图 9a 显示了在本发明的一个实施例中生成的互锁树数据仓库 900。输入中的指示符的存在，在本示例中，词组末尾或句子末尾指示符（例如，“FURRY”之后的句点），可以触发将一个级别的最终产物（**BOT-C-A-T-EOT 908**、**BOT-A-R-E-EOT 906**、**BOT-F-U-R-R-Y-EOT 904**）组合成下一个级别的子组件，即，一个级别（例如，级别 1910）最终产物节点（例如，诸如“CATS”、“ARE”和“FURRY”之类的单词）可以变成代表下一级别（例如，级别 2912）的数据集元素的根节点。因此，节点“**BOT-CATS-ARE-FURRY-EOT**”902 是代表句子“CATS ARE FURRY”的单个节点。

在本发明的一个实施例中，代表较高级别的数据集元素的节点不包含数据或数据的表示方法或概念；即，代表较高级别的数据集元素的基本根节点只包含指向较低级别中的节点的指针。例如，图 9b 显示了图 9a 的某些节点的内容。值得注意的是，节点 **BOT-C-A-T-S-EOT** 被用作级别 2912 的基本根节点（节点 908 的 **asResultList 914** 包含 300，节点 **BOS-CATS 916** 的位置，而节点 **BOS-CATS 916** 的 **asResult** 指针 918 包含 200，节点 **BOT-C-A-T-S-EOT 908** 的位置）等等。

可以生成任何合适数量的级别。例如，在文本中，级别可以代表字母、单词、句子、段落、章节、图书、图书馆等等。可以理解，虽然在示范图中显示了互锁树数据仓库的两个级别（级别 1910 和级别 2912），但是，本发明不仅限于此。可以构建任意个级别的互锁树数据仓库。因为本示例的 **universe** 是文本，即，构成单词（最终产物的一个级别）的字母的组合，本发明的此实施例中的单词的组合的结果是词组或句子（最终产物的另一个级别）。句子可以组合起来以构成段落，段落可以组合起来，以构成章节等等。

可以进一步理解，取决于输入 **universe**，最终产物可以代表单词、短语、句子等等之外的实体。仅举一个例子：如果输入是包括染色体的氨基酸的序列，一个最终产物可以代表基因或等位基因。

现在假设输入 204 包括诸如下列记录的数据记录：

5 **Bill Tuesday 40 sold PA**
 Bill Monday 103 sold NJ
 Bill Monday 100 trial PA
 Tom Monday 80 sold PA
 Tom Monday 13 trial NJ

10 在本发明的一个实施例中，数据集元素包括被诸如但不限于空白字符之类的分隔符分隔的信息的字段。在一个实施例中，数据集元素由输入派生而来，虽然可以理解，如上所述，本发明不仅限于此。到现在为止在输入数据中遇到的数据集元素是销售员姓名（**Bill** 和 **Tom**）、星期几（星期一、星期二），商品编号（40、103、100、80、
15 **13**），状态（销售，试用）和州（**PA**，**NJ**）。在本发明的一个实施例中，将从此输入中产生图 10 的互锁树数据仓库 1000。在图 10 中，由于空间原因，没有显示节点的第一部分。例如，虽然节点 1002 被标记为“**Bill**”，节点 1002 实际上代表“**BOT-Bill**”。虽然节点 1002 被标记为“**Tuesday**”，节点 1004 实际上代表“**BOT-Bill -Tuesday**”等
20 等。

从互锁树数据仓库中访问信息

在图 11 中显示了访问存储在互锁树数据仓库中的信息的方法。在步骤 1102 中，接收要从互锁树数据仓库中检索的信息的请求。可以将待检索的信息的请求转换为可以由互锁树存取器处理的形式。在
25 步骤 1104 中，访问被指出的节点。在步骤 1106 中，检索适当的 **asCaseList** 和 / 或 **asResultList**。在步骤 1108 中，沿适当的 **asCaseList** 或 **asResultList** 而行，以检索所需要的信息。在步骤 1110 中，收集并返回所请求的信息。

例如，再次请参看图 7a，包括 **asResult** 链接 328、330、332、

334、710、712、714 和 716 的数据仓库 700a 可以用来确定上下文的诸如下列问题的答案：“哪些节点包括字母 A？”，“A 在哪些字母前面或后面？”，“哪些（或多少）单词包括字母 A？”，“哪些单词既包含字母 A 又包含 T？”，“哪些单词包含 A 并且 A 前面有 T？”
5 和数不清的其他问题。（可以理解，虽然为便于理解，在示例问题中指定了特定的字母，但是可以替代任何数据集元素或单元组。）

例如，在本发明的一个实施例中，通过遵循代表数据集元素的特定节点的 `asResultList` 中包含的指针，可以判断包含所需要的数据集元素的节点和最终产物。访问 `asresultlist`，并沿列表中的每一个指针
10 而行，直到与该节点关联的 `asCase` 树枝。如果需要最终产物，则沿 `asCase` 树枝树而行，直到树枝的叶节点。

在本发明的一个实施例中，对信息的请求是指定约束和/或焦点的形式。例如，对信息的请求可以是约束的列表的形式。约束的列表可以嵌套，也可以是独立的。在本发明的一个实施例中，查找每一个
15 列出的约束的 `AsResultList`，查找每一个约束的每一个 `AsResultList` 内的每一个节点的树枝，沿树枝而行，到它们的最终产物，选择每一个约束的每一个 `asResultList` 内的每一个树枝的最终产物的交叉点。通过首先约束数据仓库以检索一组数据，然后该组数据被用作被进一步约束的数据集，来查找嵌套的约束。

在本发明的一个实施例中，焦点判断返回的信息。在其中数据集元素是字母（级别 1 最终产物包括单词，级别 2 包括句子）两级别
20 数据仓库的情况下，指定的约束是特定字母，指定焦点为“单词”将导致只返回单词，指定焦点为“句子”将导致只返回句子。从第一级别检索最终产物将导致返回单词。从第二级别检索最终产物将导致返回句子。在一个实施例中，为检索句子，沿每一个单词的 `asResultList` 而
25 行，直到下一个级别，并沿指定的树枝而行，直到其最终产物，以检索包括指定的字母的句子。

在一个实施例中，通过沿具有级别开始指示符作为其根的树而行，可以查找以约束开始的所有最终产物（例如，可以查找以指定字

母开始的所有单词。)同样,可以查找具有指定约束的所有最终产物,或指定位置中的指定约束(例如,具有特定字母的所有单词或在指定列中具有指定字母的所有单词)。同样,通过沿具有级别结束指示符作为根的树而行,可以查找以指定约束结束的所有最终产物(例如,以指定字母结束的所有单词。)可以指定多个约束和/或焦点。

例如,假设需要包含诸如字母(例如,字母“B”)之类的数据集元素的树的 asCase 树枝的第一个节点。在一个实施例中,从存储器中检索代表数据元素的基本根节点(例如,节点 B 718),访问其 asResultList (例如, asResultList 742f),以返回通过某些子组件与基本根节点(例如,节点 B 718)组合而创建的节点的位置。访问 asResultList 中的节点。在本示例中,访问位置 180,该位置保留了节点 BOT-T-A-B 707。如此,节点 BOT-T-A-B 707 是互锁树数据仓库 700a 中的包括字母“B”的表示法的节点。为查找形成的最终产物,通过反复检索被访问的节点的 asCaseList,直到被检索到的 asCaseList 是空值,沿着 asCase 树枝而行(例如,在本示例中,包含节点 BOT-T 703、节点 BOT-T-A 705、节点 BOT-T-A-B 707 和节点 BOT-T-A-B-EOT 709 的树枝)。例如,为判断包含数据集元素 B 718 的单词是“TAB”,访问节点 BOT-T-A-B 707 的 asCaseList 740f,以检索位置 185。访问位置 185 的内容,以检索 asCaseList 748f。由于 asCaseList 748f 是 Null 指针,因此,已经到达最终产物。

仍请参看图 7a,现在假设需要包含字母“A”的所有 asCase 树枝的第一个节点。从存储器中检索基本根节点 A 304,并访问其 asResultList 542f,以返回位置 145 和 175。访问包含节点 BOT-C-A 314 的第一位置 145。节点 BOT-C-A 314 是数据结构 700a 的包括字母“A”的第一树枝中的第一节点。为查找形成的最终产物,通过反复检索节点的 asCaseList,直到被检索到的 asCaseList 是空值,沿着 asCase 树枝(例如,在本示例中,包含节点 BOT-C 312、节点 BOT-C-A 314、节点 BOT-C-A-T 316 和节点 BOT-C-A-T-EOT 318

的树枝)、树枝的 asCase 链接而行。例如,为判断包含数据集元素 A 304 的单词是“CAT”,访问节点 BOT-C-A 314 的 asCaseList 740f,以检索位置 145。访问位置 145(节点 BOT-C-A 314)的内容,以检索 asCaseList 532f, 150。访问位置 145(节点 BOT-C-A 314) 5 的内容,以检索 asCaseList 548f, 155。访问位置 155(节点 BOT-C-A-T-EOT 318)的内容,以检索 asCaseList 572f。由于 asCaseList 572f 是 Null 指针,因此,已经到达最终产物。

访问包含节点 BOT-T-A 705 的下一个位置 175。节点 BOT-T-A 705 是互锁树数据仓库 700a 的包括字母“A”的第二树枝 10 中的第一节点。为查找形成的最终产物,通过反复检索节点的 asCaseList,直到被检索到的 asCaseList 是空值,沿着 asCase 树枝(例如,在本示例中,包含节点 BOT-T 703、节点 BOT-T-A 705、节点 BOT-T-A-B 707 和节点 BOT-T-A-B-EOT 709 树枝),树枝的 asCase 链接而行。例如,为判断包含数据集元素 A 304 的第二个单 15 词单词是“TAB”,访问节点 BOT-T-A-B 707 的 asCaseList 740f,以检索位置 185。访问位置 185 的内容,以检索 asCaseList 748f。由于 asCaseList 748f 是 Null 指针,因此,已经到达最终产物。

再次请参看图 7a,现在假设需要包含字母“A”和“T”的所有 asCase 树枝的第一个节点。如前面所描述的,从存储器中检索基本 20 根节点 A 304,并访问其 asResultList 542f,以返回位置 145 和 175。访问包含节点 BOT-C-A 314 的第一位置 145。节点 BOT-C-A 314 是互锁树数据仓库 700a 的包括字母“A”的第一树枝中的第一节点。为查找形成的最终产物,通过反复检索节点的 asCaseList,直到被检索到的 asCaseList 是空值,沿着 asCase 树枝(例如,在本示 25 例中,包含节点 BOT-C 312、节点 BOT-C-A 314、节点 BOT-C-A-T 316 和节点 BOT-C-A-T-EOT 318 的树枝)、树枝的 asCase 链接而行。例如,为判断包含数据集元素 A 304 的单词是“CAT”,访问节点 BOT-C-A 314 的 asCaseList 740f,以检索位置 145。访问位置 145(节点 BOT-C-A 314)的内容,以检索 asCaseList 532f, 150。

访问位置 145(节点 BOT-C-A 314)的内容,以检索 asCaseList 548f, 155。访问位置 155(节点 BOT-C-A-T-EOT 318)的内容,以检索 asCaseList 572f。由于 asCaseList 572f 是 Null 指针,因此,已经到达最终产物。最终产物节点 BOT-C-A-T-EOT 318 包含数据集元素 A。

访问包含节点 BOT-T-A 705 的下一个位置 175。节点 BOT-T-A 705 是互锁树数据仓库 700a 的包括字母“A”的第二树枝中的第一节点。为查找形成的最终产物,通过反复检索节点的 asCaseList,直到被检索到的 asCaseList 是空值,沿着 asCase 树枝 (例如,在本示例中,包含节点 BOT-T 703、节点 BOT-T-A 705、节点 BOT-T-A-B 707 和节点 BOT-T-A-B-EOT 709 树枝),树枝的 asCase 链接而行。例如,为判断包含数据集元素 A 304 的第二个单词单词是“TAB”,访问节点 BOT-T-A-B 707 的 asCaseList 740f,以检索位置 185。访问位置 185 的内容,以检索 asCaseList 748f。由于 asCaseList 748f 是 Null 指针,因此,已经到达最终产物。最终产物节点 BOT-T-A-B-EOT 709 包含数据集元素 A。

接下来,从存储器中检索基本根节点 T 308,并访问其 asResultList 558f,以返回位置 150 和 170。访问包含节点 BOT-C-A-T 316 的第一位置 150。节点 BOT-C-A-T 316 是互锁树数据仓库 700a 的包括字母“T”的第一树枝中的第一节点。为查找形成的最终产物,通过反复检索节点的 asCaseList,直到被检索到的 asCaseList 是空值,沿着 asCase 树枝(例如,在本示例中,包含节点 BOT-C 312、节点 BOT-C-A 314、节点 BOT-C-A-T 316 和节点 BOT-C-A-T-EOT 318 的树枝)、树枝的 asCase 链接而行。例如,为判断包含不可分的基本单元 T 308 的单词是“CAT”,访问节点 BOT-C-A 314 的 asCaseList 532f,以检索位置 145。访问位置 145(节点 BOT-C-A 314)的内容,以检索 asCaseList 532f, 150。访问位置 145(节点 BOT-C-A 314)的内容,以检索 asCaseList 548f, 155。访问位置 155(节点 BOT-C-A-T-EOT 318)的内容,以检索

asCaseList 572f。由于 **asCaseList 572f** 是 Null 指针，因此，已经到达最终产物。最终产物节点 **BOT-C-A-T-EOT 318** 包含数据集元素 **T**。

访问包含节点 **BOT-T 703** 的下一个位置 **170**。节点 **BOT-T 703** 是互锁树数据仓库 **700a** 的包括字母“T”的第二树枝中的第一节点。为查找形成的最终产物，通过反复检索节点的 **asCaseList**，直到被检索到的 **asCaseList** 是空值，沿着 **asCase** 树枝（例如，在本示例中，包含节点 **BOT-T 703**、节点 **BOT-T-A 705**、节点 **BOT-T-A-B 707** 和节点 **BOT-T-A-B-EOT 709** 树枝），树枝的 **asCase** 链接而行。

例如，为判断包含数据集元素 **T 308** 的第二个单词单词是“TAB”，访问节点 **BOT-T-A-B 707** 的 **asCaseList 740f**，以检索位置 **185**。访问位置 **185** 的内容，以检索 **asCaseList 748f**。由于 **asCaseList 748f** 是 Null 指针，因此，已经到达最终产物。最终产物节点 **BOT-T-A-B-EOT 709** 包含数据集元素 **T**。如此，包含 **A** 和 **T** 的最终产物包括包含 **A** 的最终产物集与包含 **T** 的最终产物集的交集，或，在此情况下：**BOT-C-A-T-EOT 318** 和 **BOT-T-A-B-EOT 709**。

在本发明的一个实施例中，显示或打印检索到的信息。为显示或打印检索到的信息，从最终产物到开始 (**BOT**) 反向沿 **asCase** 树而行。在沿着 **asCase** 树的每一个节点中，结果指针（指向从中派生了节点的第二个部分）用来判断基本根节点代表了什么。如果互锁树数据仓库包括一个以上的级别，则结果指针指向较低级别的最终产物，必须执行相同的过程，直到检索到最低级别的基本根节点。

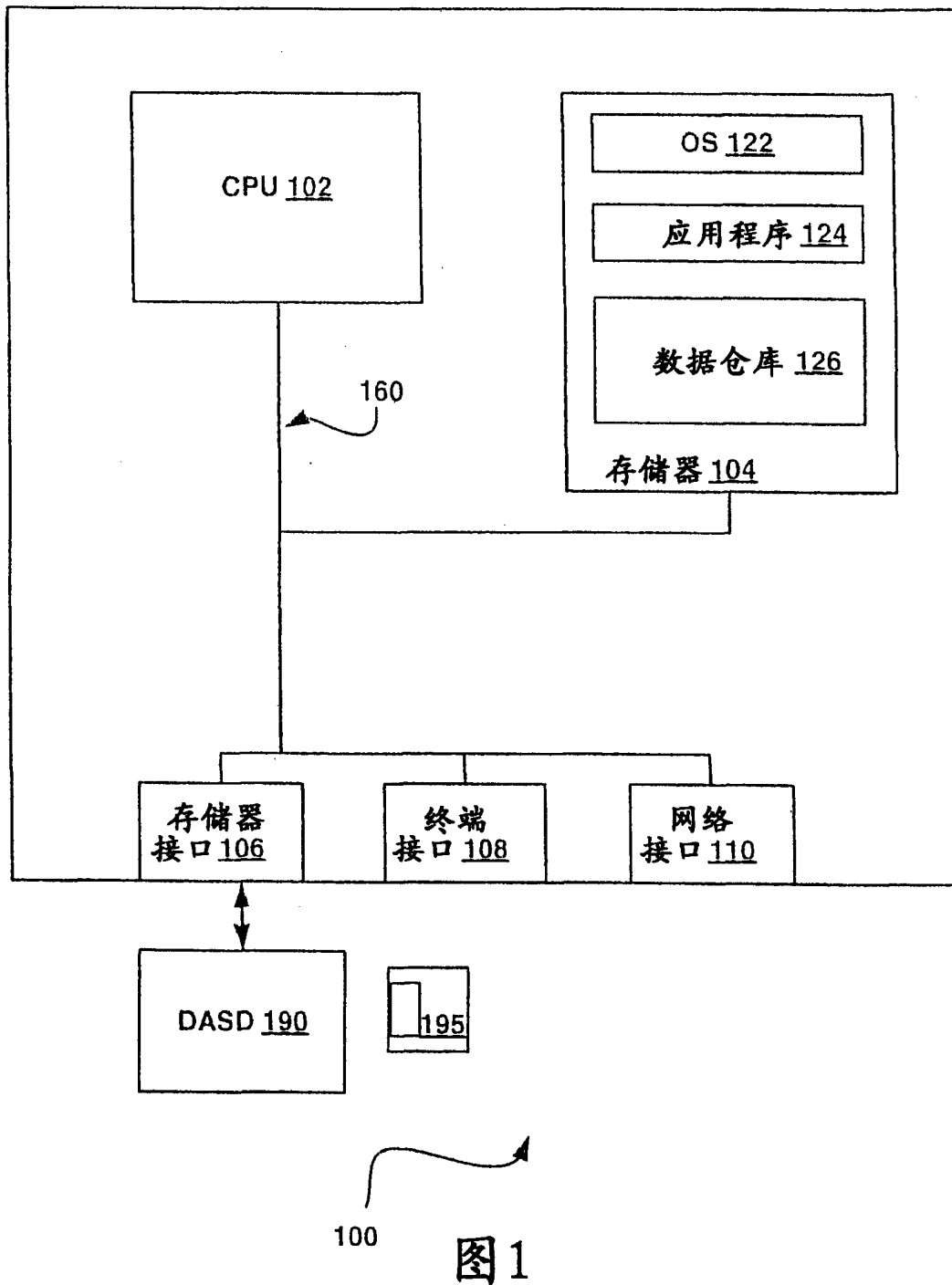
现在请参看图 10，假设需要星期二销售的物品总数。对那些精通本技术的人员显而易见的，不是遍历整个数据仓库中的所有节点，在本发明的一个实施例中，要检索此信息，只需要检索基本根节点 **1006** 和 **1008** 的 **asResultList**。遍历树枝 **5 1010**，因为代表星期二 **1006** 的基本节点指向节点 **1004**，并且因为代表销售 **1008** 的基本节点指向节点 **1026**，遍历树枝 **4 1012**，因为代表销售 **1008** 的基本节点指向节点 **1028**。不必遍历树枝 **1 1015**、**2 1014** 和 **3 1013**。从沿基

本节点 1006 和 1008 指向的树枝而行所返回的最终产物的交集包括代表 Bill Tuesday 40 sold PA 的节点 1030。

5 销售的物品数量可以通过沿从节点 1024 到代表数量 40（未显示）的根节点的指针而行来确定。可以理解，此步骤可以在发现最终产物的交叉点之后，或者，也可以在遍历树枝时检索和存储此信息。

上文所描述的方法和系统可以以存储在诸如软盘、CD-ROM、DVD-ROM、DVD-RAM、硬盘驱动器之类的计算机可读的介质或任何其他机器可读的存储介质上的程序代码（即，指令）的形式来实现，其中，当程序代码被加载到诸如计算机之类的机器中并由机器执行
10 时，机器变成实践本发明的设备。本发明也可以以通过某些传输介质（如通过电线或电缆，通过光纤，通过网络，包括因特网或内部网，或任何其他形式的传输）传输的程序代码的形式来实现，其中，当接收到程序代码并加载到诸如计算机之类的机器中并由机器执行时，机器变成实践本发明的设备。当在通用处理器上实现时，程序代码与处理
15 器结合，以提供与特定逻辑电路类似地工作的唯一设备。程序代码可以以诸如 C、C++ 或 Java 之类的高级程序设计语言来实现。或者，程序代码也可以以汇编或机器语言来实现。在任何情况下，语言都可以是编译的或解释的语言。

值得注意的是，前面所提供的示例只是为了说明，决不能理解为本发明
20 对本发明的限制。例如，互锁树数据仓库可以使用面向对象的技术、过程技术、这些技术的混合技术或任何其他合适的方法来实现。此外，虽然所提供的示例显示了存储在存储器中的数据集元素，那些精通本技术的人员将理解，此功能可以以许多不同的方式来实现。例如，本发明也使用存储在多个远程机器上的许多不同的 universe 的数据集
25 元素的许多不同集。



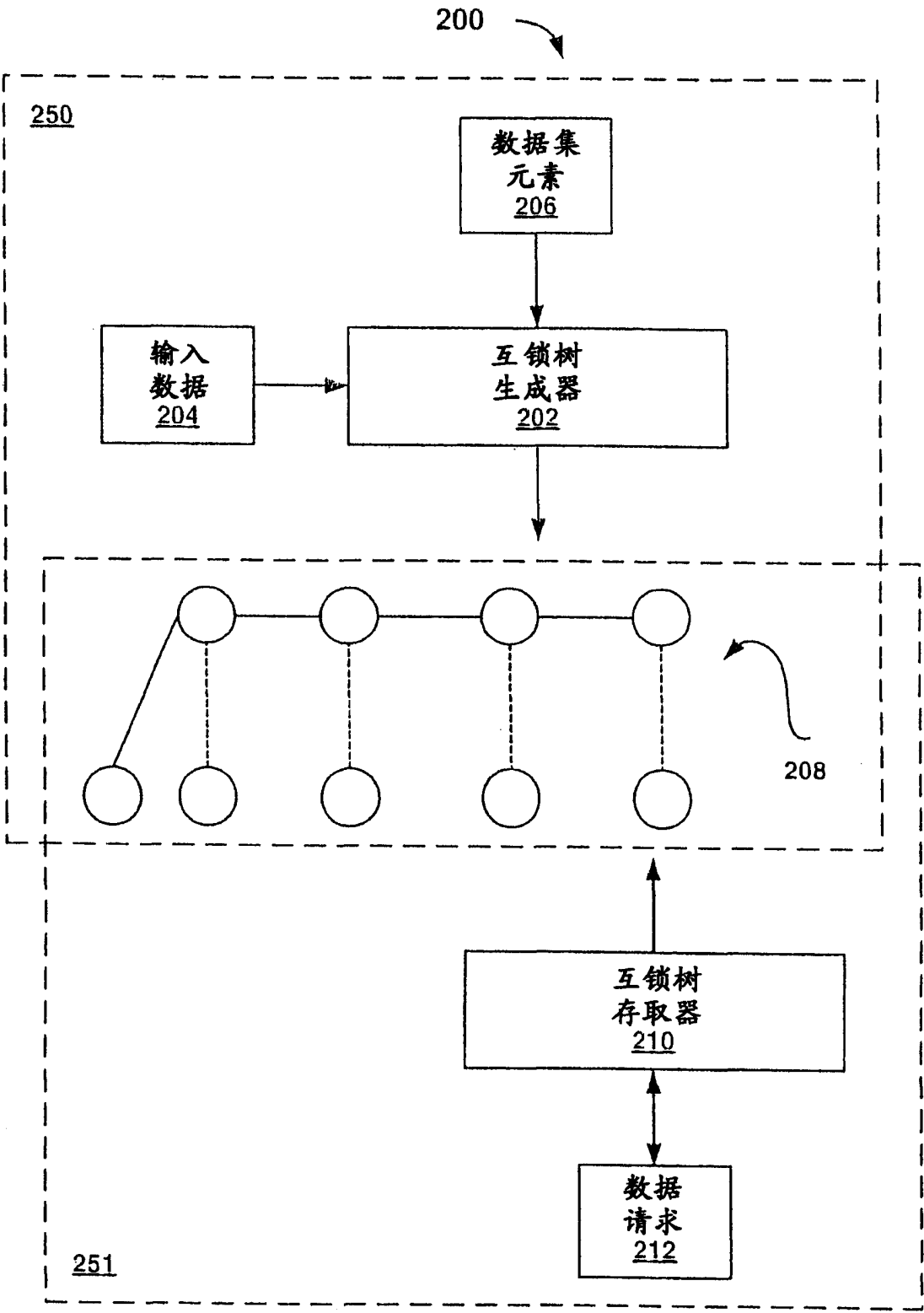


图 2a

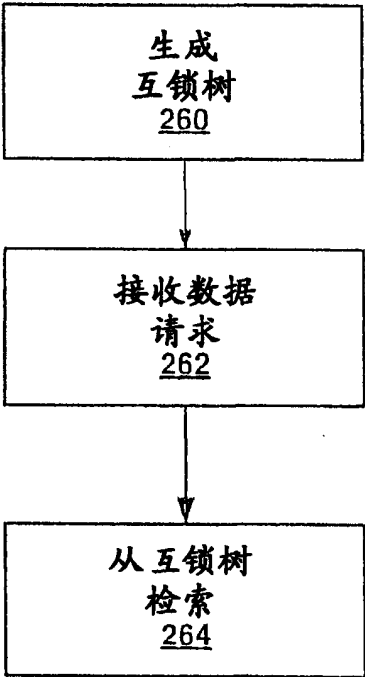


图 2b

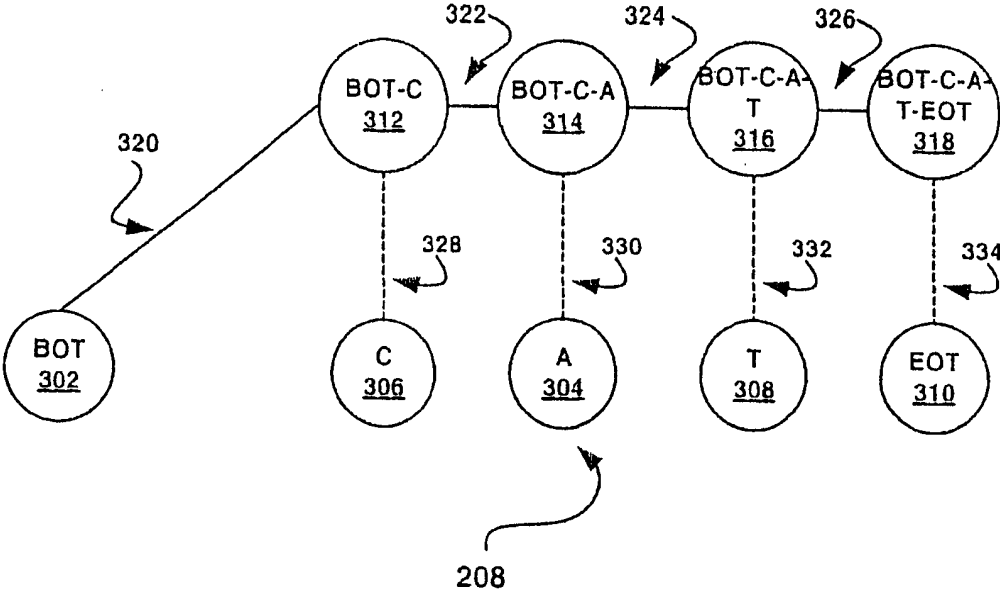


图 3a

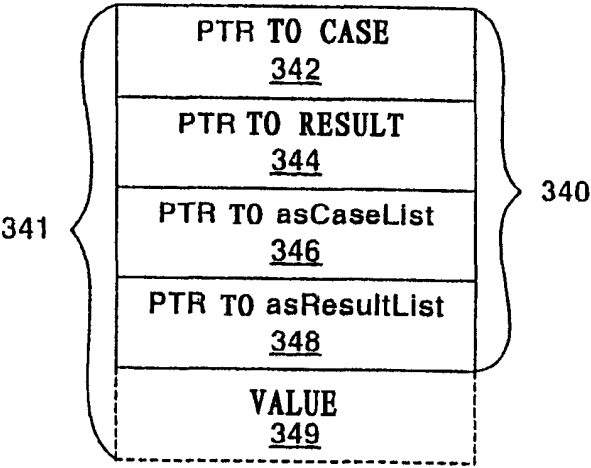


图 3b

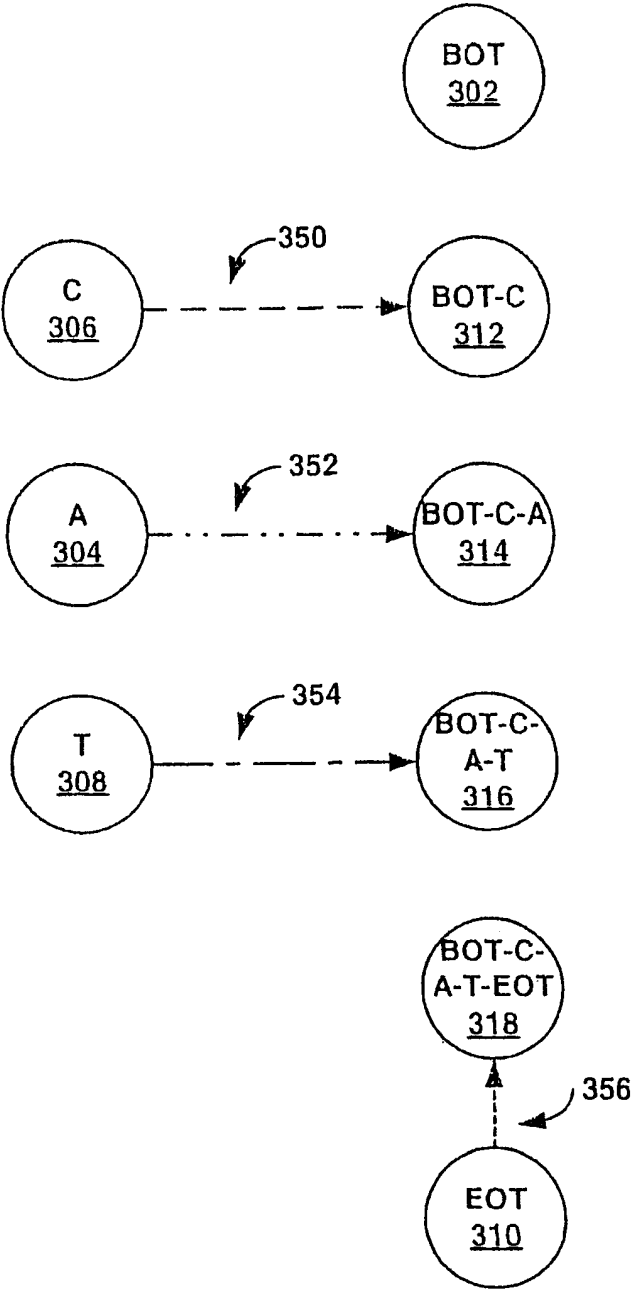


图 3c

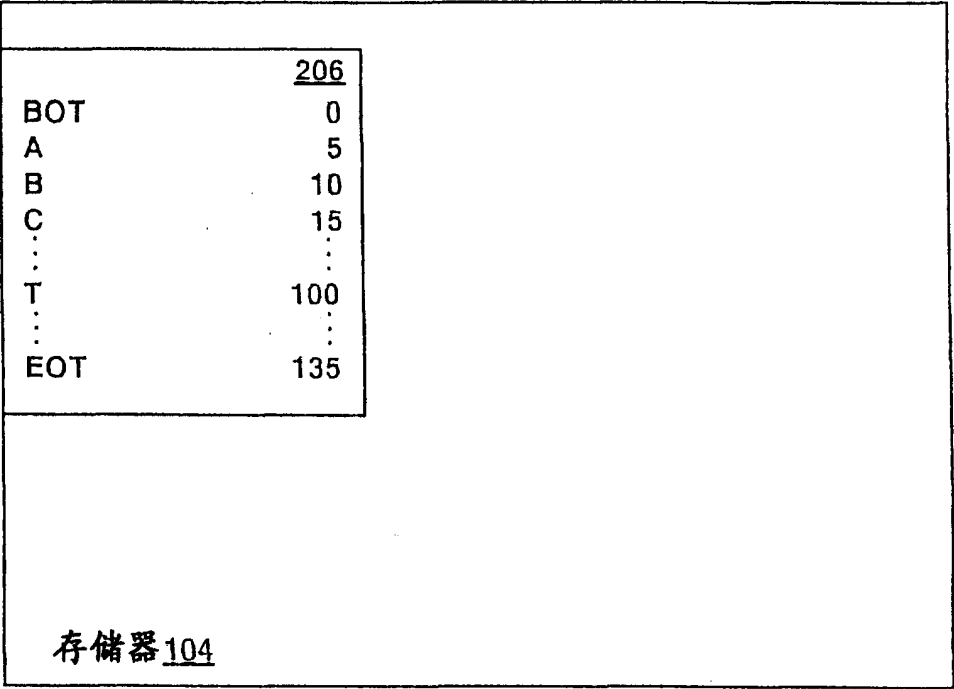


图 4

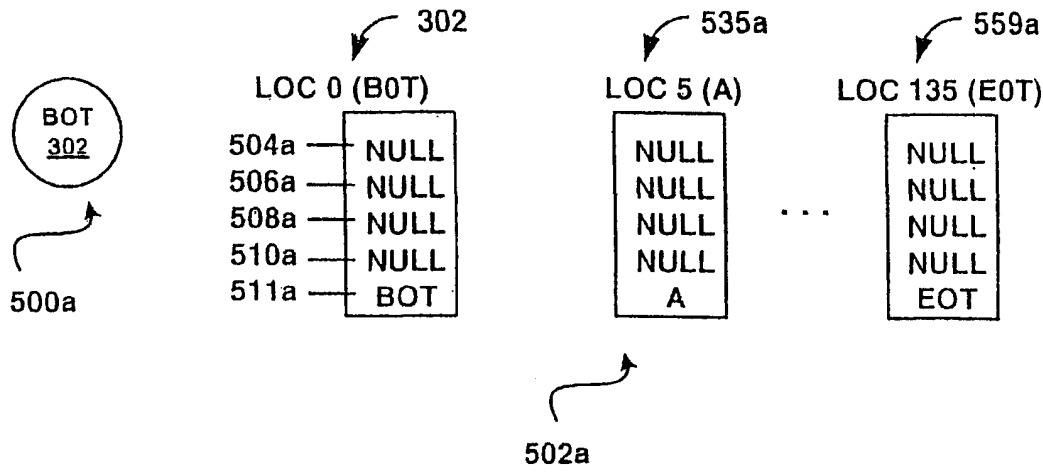


图 5a

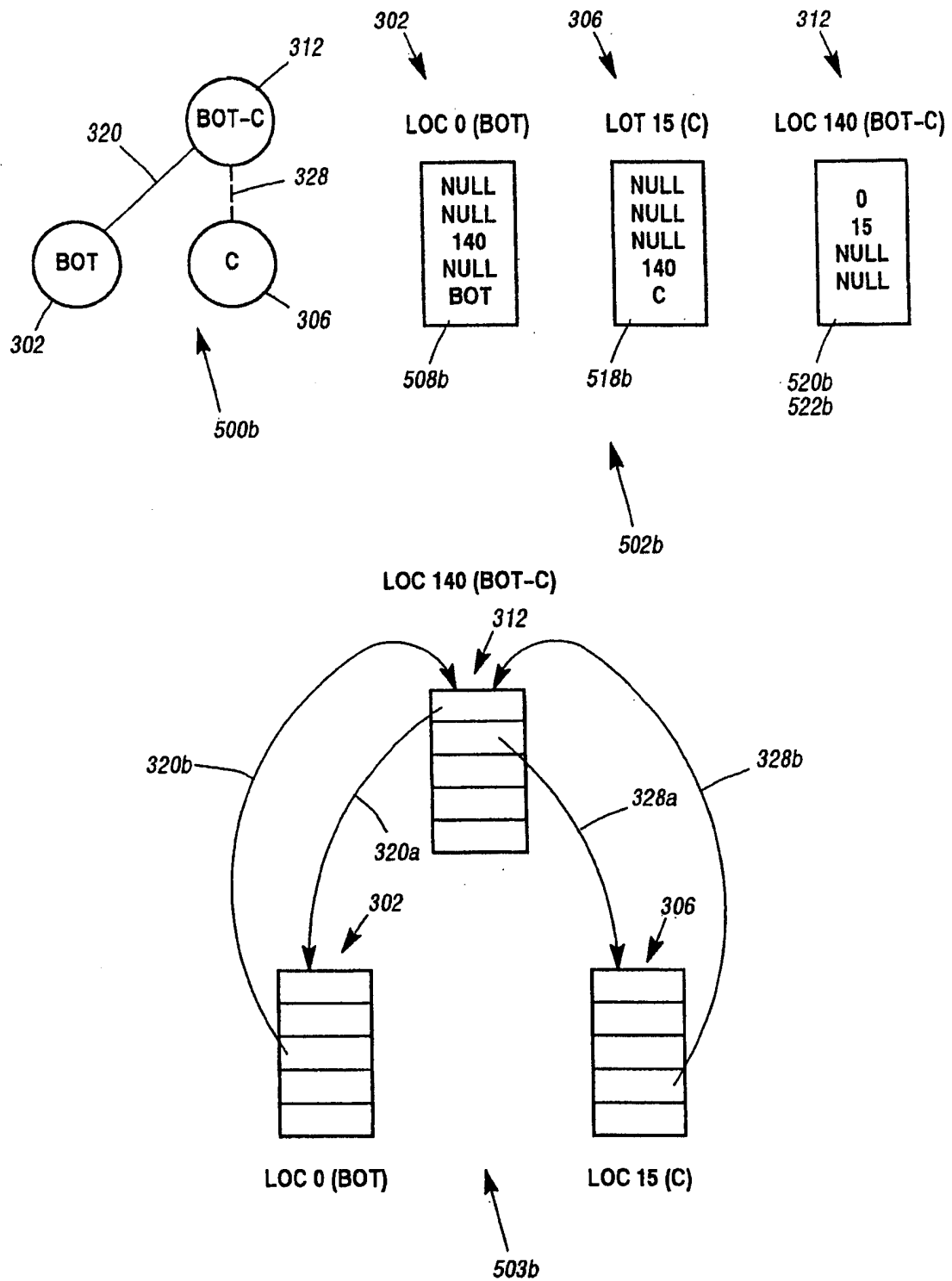


图 5b

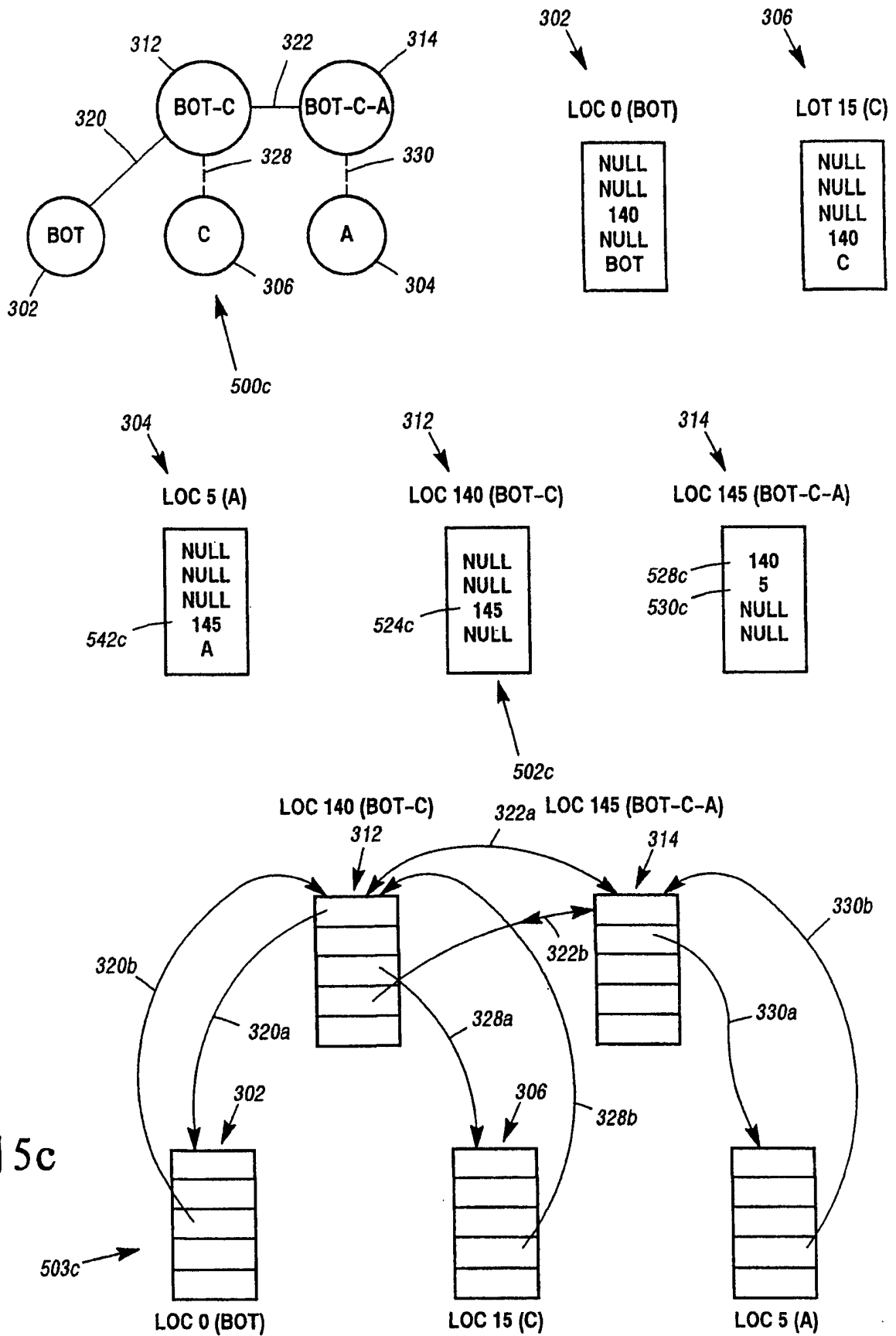
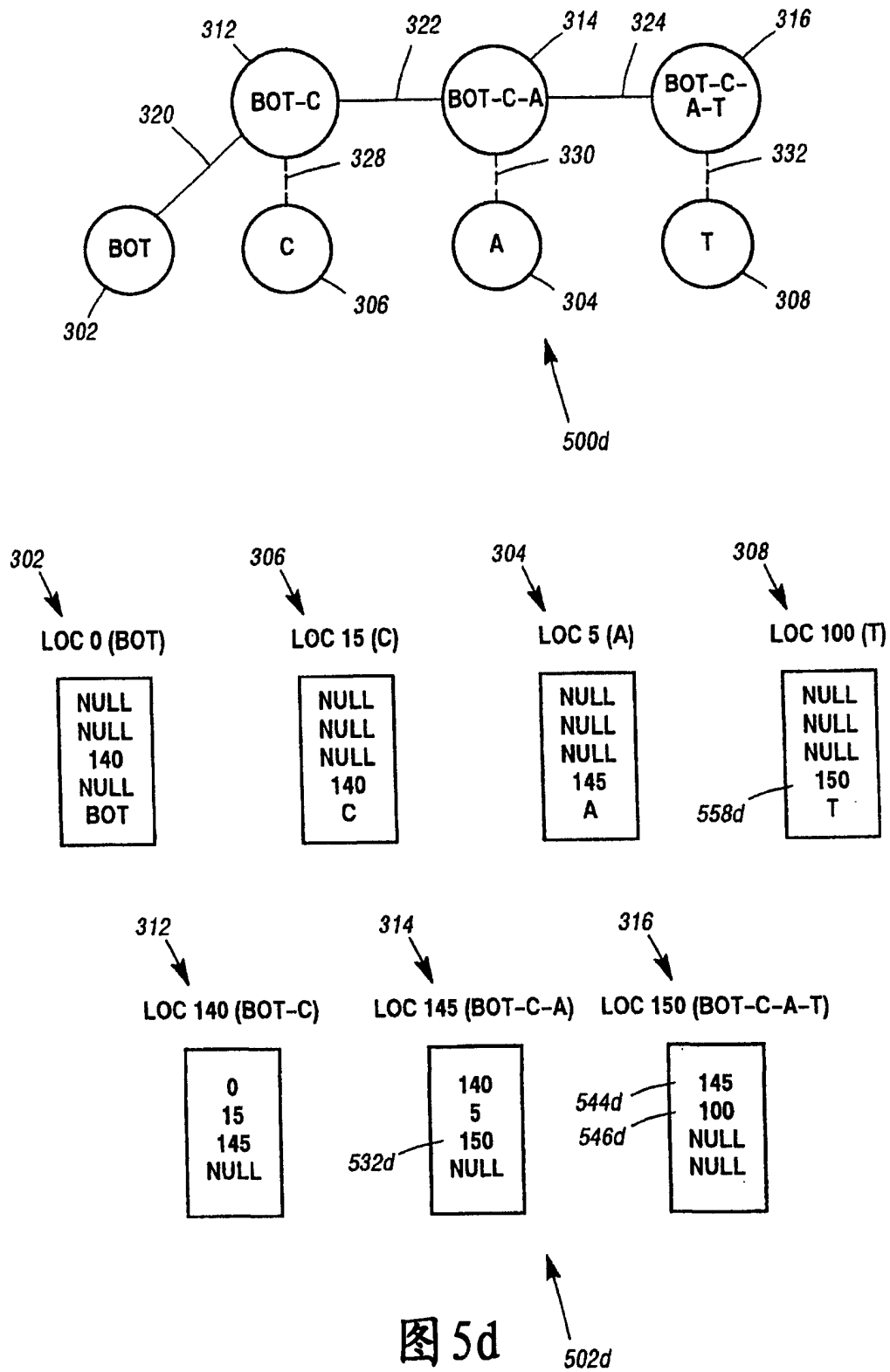
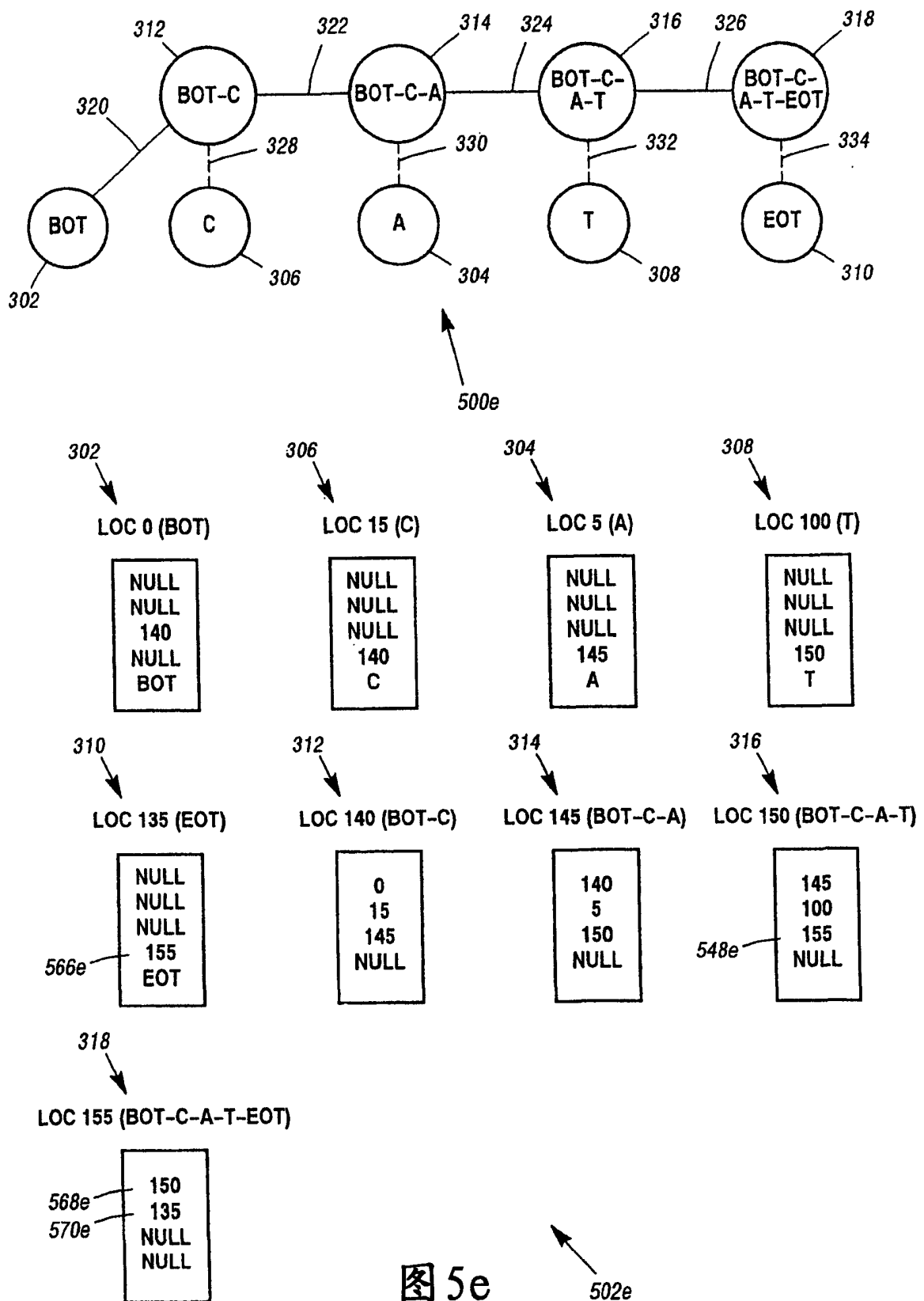


图 5c





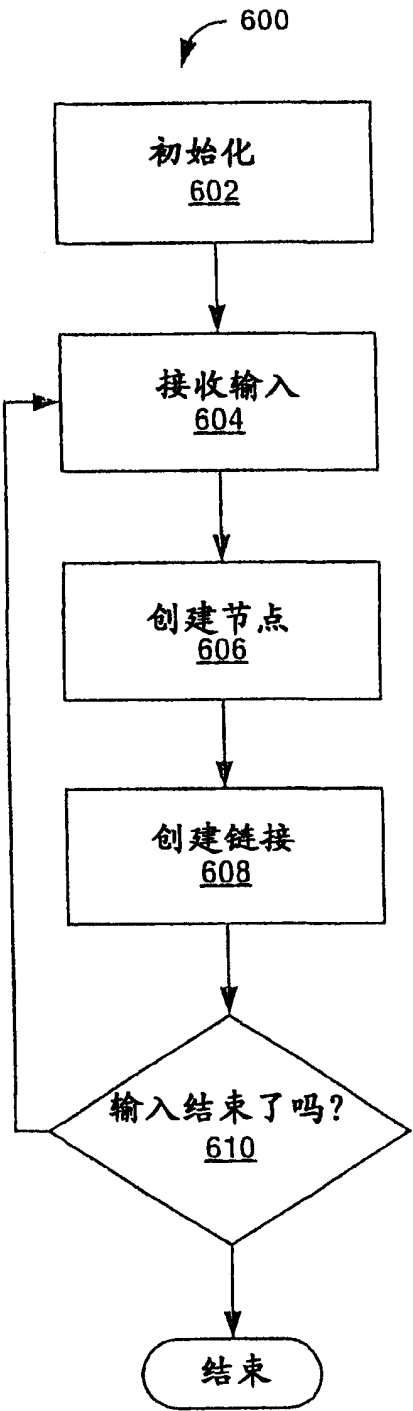


图 6

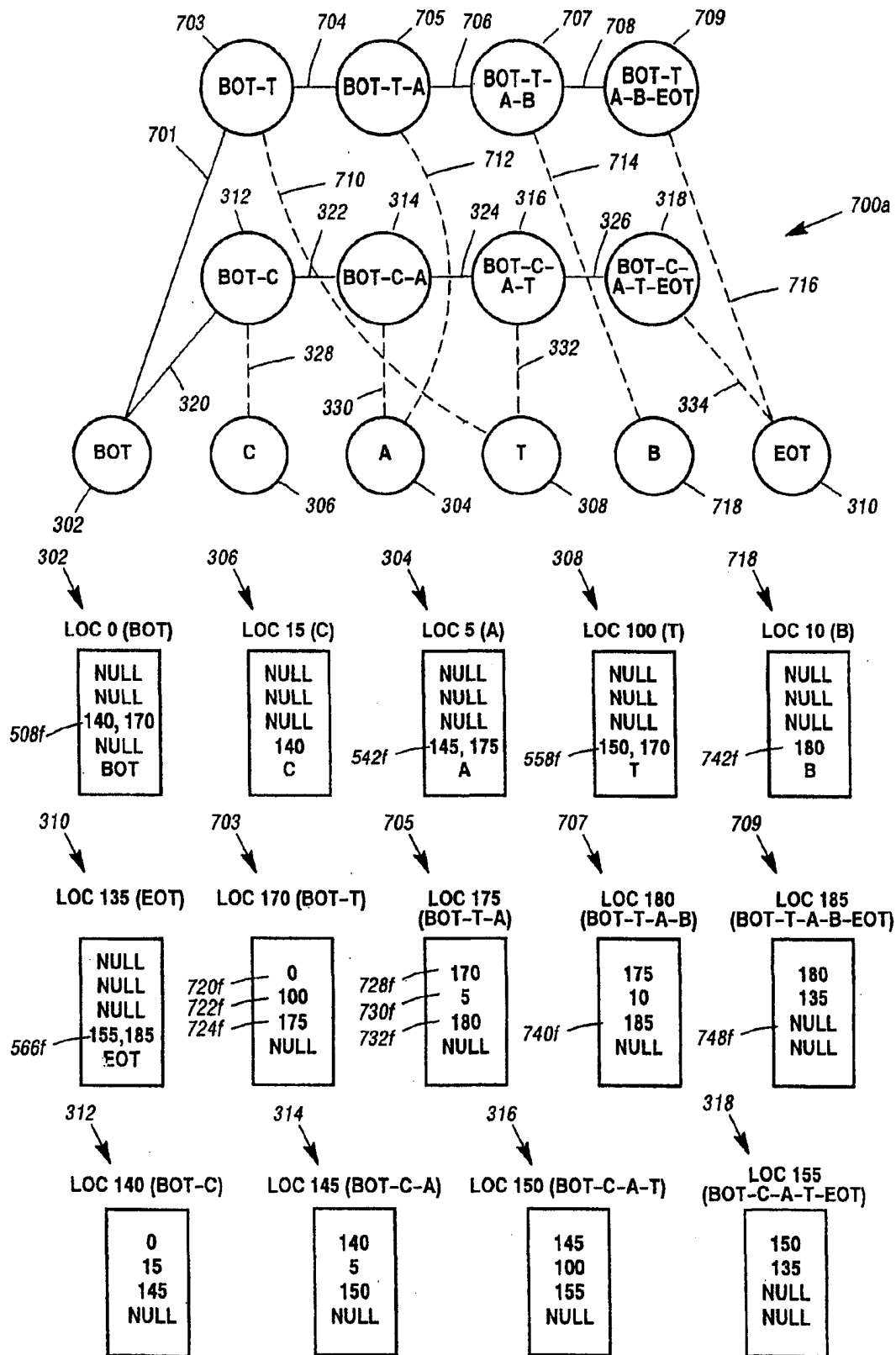


图 7a

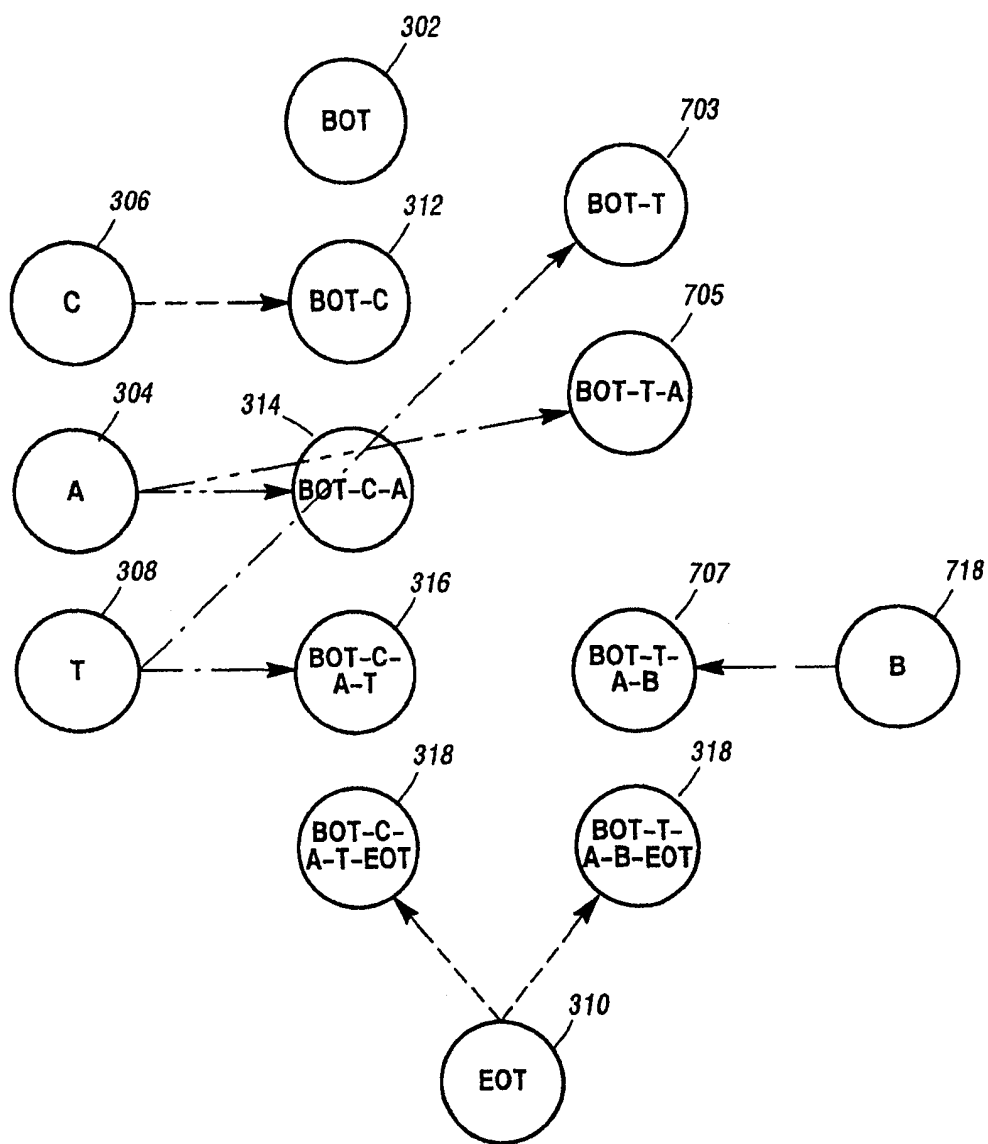


图7b

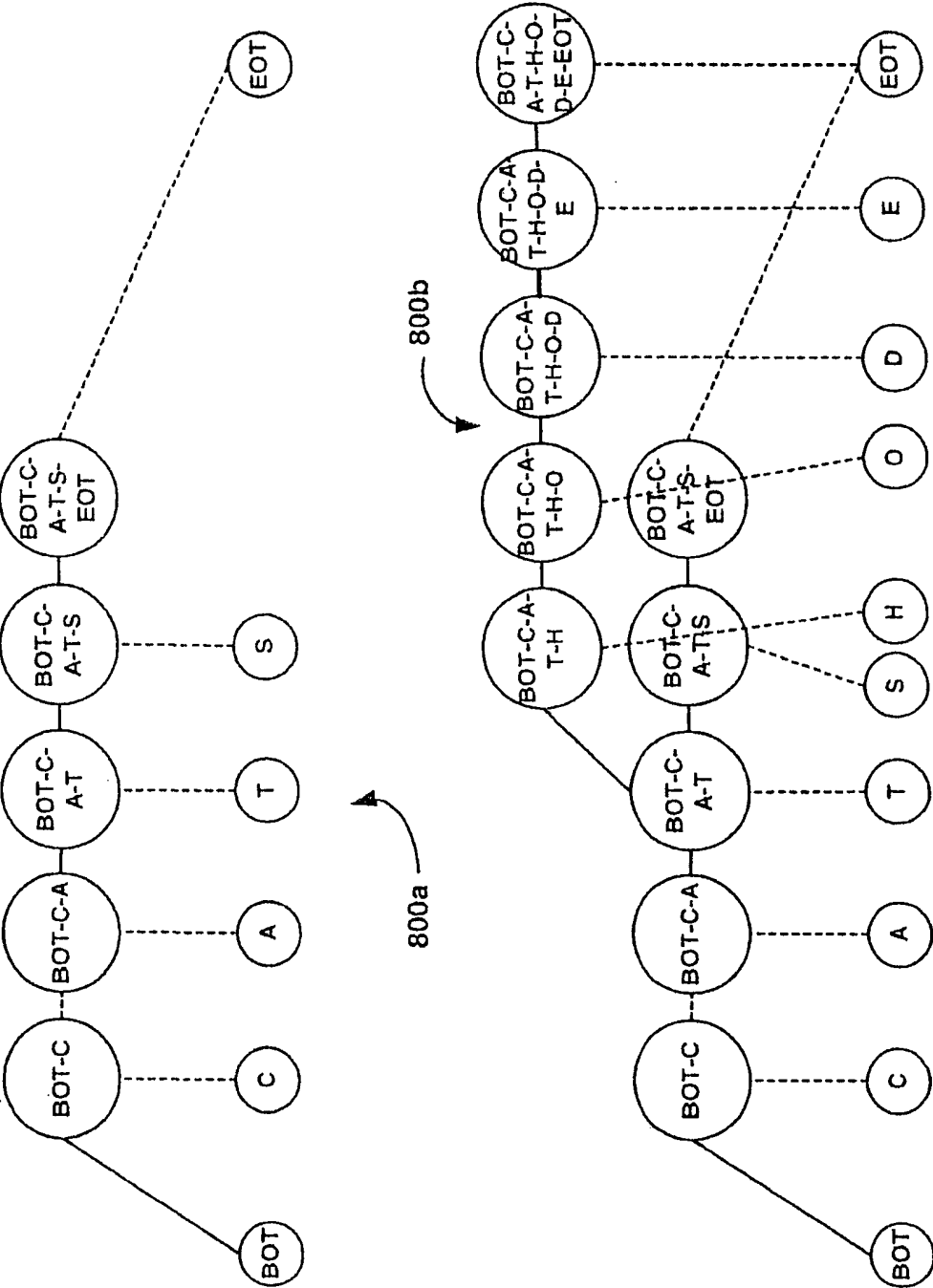


图8

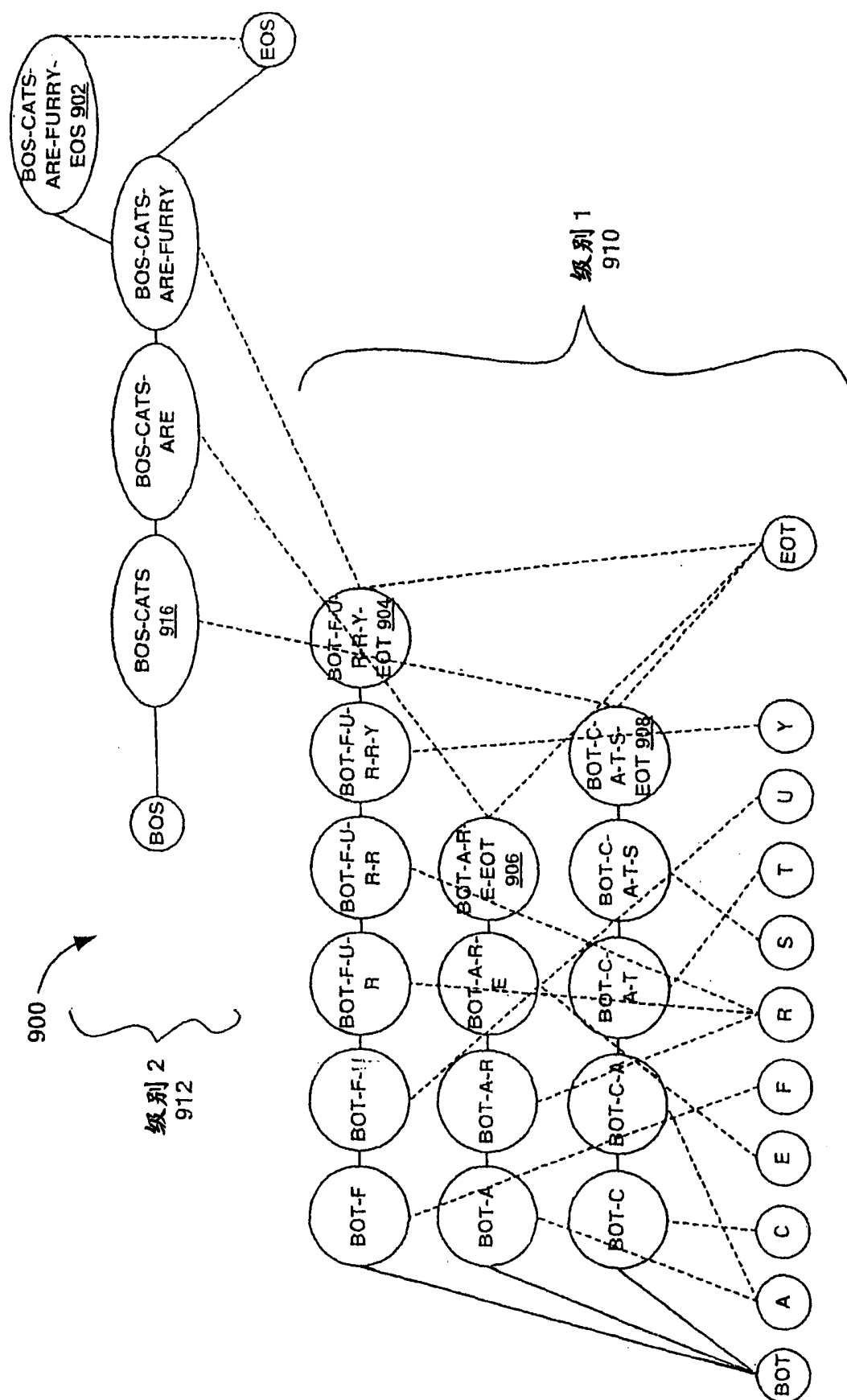


图 9a

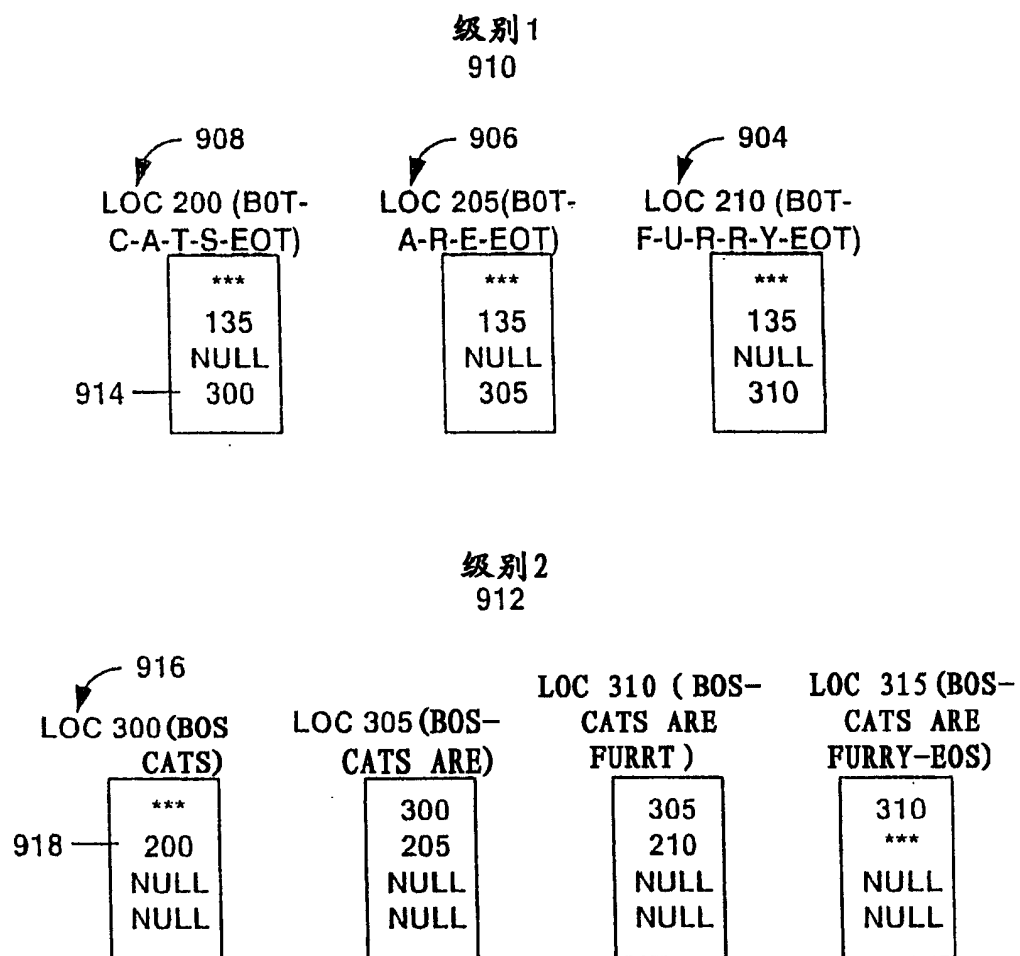


图9b

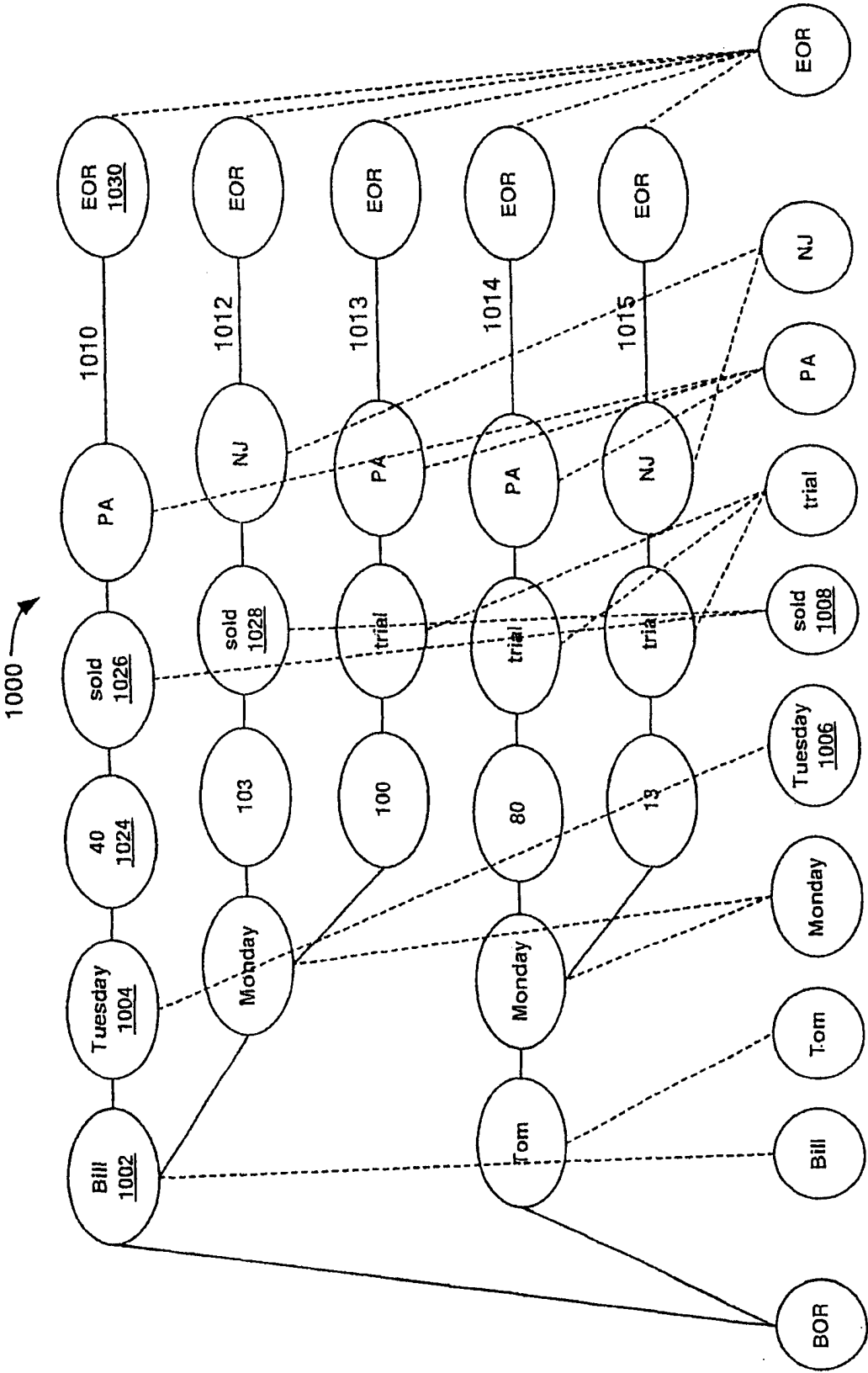


图10

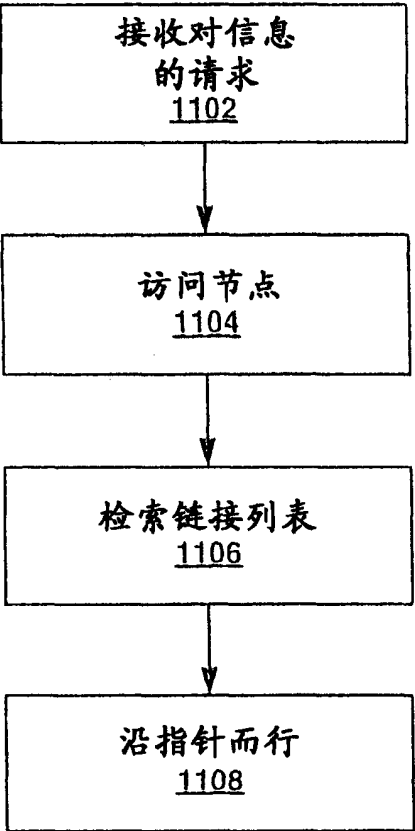


图 11