arXiv:2305.01721v1 [cs.AI] 2 May 2023

从决策规则系统构造决策树和无环决策图

Kerven Durdymyradov 和 Mikhail Moshkov计算机、电气与数学科学与工程部及计算生物科学研究中心阿卜杜拉国王科技大学（KAUST）沙特阿拉伯，图瓦勒，邮编 23955-6900{kerven.durdymyradov,mikhail.moshkov}@kaust.edu.sa

摘要

决策树和决策规则系统被广泛用作分类器、知识表示的手段以及算法。它们是数据分析中最具可解释性的模型之一。研究这两种模型之间的关系可以被视为计算机科学的一项重要任务。将决策树转换为决策规则系统的方法简单且广为人知。本文考虑逆变换问题，该问题并不简单。我们研究从决策规则系统构建表示决策树的决策树和无环决策图的复杂性，并讨论不构建整个决策树的可能性，而是针对给定输入描述该树中的计算路径。

关键词：决策规则系统、决策树、无环决策图。

1 引言

本文探讨将决策规则系统转换为决策树的问题。本文基于我们先前的工作 [12]。在该工作中，我们证明了，由决策规则系统构造出的决策树，其最小深度可能远小于该系统规则中不同属性的数量。在这种情况下，使用决策树是合理的。

在本文中，对于某些类型的决策规则系统和问题，我们证明了构造决策树以及表示决策树的两类无环决策图存在多项式时间算法。在其他所有情况下，我们利用以下事实证明不存在此类算法：决策树或无环决策图中的最小节点数可能会随着决策规则系统大小的变化而呈超多项式函数增长。为避免决策树中节点数量所带来的困难，我们还讨论了不构建完整决策树，而是针对给定输入描述该树中计算路径的可能性。

决策树 [3, 4, 8, 30, 33, 39]和决策规则系统 [6, 7, 11, 13, 32, 33, 34, 35]被广泛用作分类器、知识表示的手段以及算法。它们是数据分析 [10, 14, 22, 40]中最具可解释性的模型之一。

研究这两种模型之间的关系，可以视为计算机科学的一项重要任务。将决策树转换为决策规则系统的方法简单且广为人知 [36, 37, 38]。本文考虑的是逆变换问题，这并非易事。

与这一问题相关的主要研究方向如下：

- 决策树的两阶段构建。首先，根据输入数据构建决策规则，然后根据构建好的规则生成决策树或决策结构（决策树的推广形式）。有关详情（包括对这一方法优点的说明），请参见 [1, 2, 16, 17, 18, 19, 20, 21, 41]。

- 用于计算布尔函数的确定性与非确定性决策树深度之间的关系 [5, 15, 23, 42]。请注意，非确定性决策树可以解释为决策规则系统。计算布尔函数的非确定性决策树的最小深度等于其证书复杂度 [9]。

- 有限与无限信息系统上问题的确定性与非确定性决策树深度之间的关系；每个信息系统均由一个论域及其上定义的一组属性构成 [24, 26, 28, 29, 31]。

本文继续发展所谓的句法方法，以研究文献 [25, 27] 中提出的所考虑问题。该方法假定我们不知道输入数据，而只有一个必须转换为决策树的决策规则系统。

设有一个如下形式的决策规则系统 S

$$(a_{i_1} = \delta_1) \wedge \cdots \wedge (a_{i_m} = \delta_m) \to \sigma,$$

其中 a_{i_1}、.....、 a_{i_m} 为属性， δ_1、.....、 δ_m 为这些属性的取值，而 σ 为一个决策。我们描述与该系统相关的三个问题：

- 对于给定的输入（ S中包含的所有属性的取值组成的元组），需要找出所有对该输入可实现的规则（即左侧为真的规则），或者证明不存在这样的规则。

- 对于给定的输入，需要找出所有可由该输入实现的规则右侧，或者证明不存在这样的规则。

- 对于给定的输入，需要找出至少一条可由该输入实现的规则，或者证明不存在这样的规则。

对于每个问题，我们考虑其两种变体。第一种假设输入中的每个属性只能取系统 S 中该属性实际出现过的值。第二种情况下，我们假设输入中的任何属性都可以取任意值。

我们的目标是尽量减少对属性值的查询次数。为此，我们将决策树作为解决所考虑的六个问题的算法进行研究。

对于这些问题中的每一个，我们研究了关于决策规则系统三个参数的 [12]不可改进的上下界：系统中规则所包含的不同属性总数、决策规则的最大长度，以及属性值的最大数量。

我们证明，对于每个问题，都存在一些决策规则系统，使得能够解决该问题的决策树的最小深度远小于规则系统中的属性总数。对于这类决策规则系统，使用决策树是合理的。

在本文中，对于所考虑的问题中的每一个，我们研究

- 构造表示决策树的决策树和无环决策图的复杂性。

- 不构建完整决策树，而是针对给定输入描述该树中计算路径的可能性。

我们证明，在许多情况下，决策树中的节点最小数量会随着决策规则系统的大小呈超多项式函数增长。我们表明，可以使用两类表示决策树的无环决策图来解决这一问题。然而，在这种情况下，需要同时最小化无环图的深度和节点数量，这是一个困难的双准则优化问题。我们将这一问题留待今后研究，并研究一种处理决策树的不同方法：不构建完整的决策树，而是使用多项式时间算法，对给定属性值元组上的决策树运行过程进行建模。

本文重复了 [12] 中的主要定义，并给出 [12] 中的一些引理，但不附证明。在备注3和备注4（见第7节）中，我们提及了一些已发表于 [27] 的结果，这些结果在本文中未附证明，并进行了详细讨论。

本文共分为八节。第2节讨论主要定义和记号。第3—6节探讨构造表示决策树的决策树和无环决策图的问题。第7节讨论一种可能性：对于给定输入，不构造整棵决策树，而只构造该树中的计算路径。第8节为简短的结论。

2 主要定义和记号

本节讨论与决策规则系统和决策树相关的主要定义和记号。事实上，我们重复了 [12]，中的定义和记号，但考虑了新的示例。

2.1 决策规则系统

令 $\omega = \{0, 1, 2, \ldots\}$ 和 $A = \{a_i : i \in \omega\}$。集合 A 中的元素称为属性。

定义 1。决策规则是如下形式的表达式

$$(a_{i_1} = \delta_1) \wedge \cdots \wedge (a_{i_m} = \delta_m) \to \sigma,$$

其中，$m \in \omega$, $a_{i_1}, \ldots, a_{i_m}$ 是来自 A 且两两不同的属性，$\delta_1, \ldots, \delta_m, \sigma \in \omega$。

我们用 r 表示这条决策规则。表达式 $(a_{i_1} = \delta_1) \wedge \cdots \wedge (a_{i_m} = \delta_m)$ 称为规则的左侧，数值 σ 称为规则 r 的右侧。数值 m 称为决策规则 r 的长度。记 $A(r) = \{a_{i_1}, \ldots, a_{i_m}\}$ 和 $K(r) = \{a_{i_1} = \delta_1, \ldots, a_{i_m} = \delta_m\}$。若 $m = 0$，则 $A(r) = K(r) = \emptyset$。

定义 2。若 $K(r_1) = K(r_2)$，且规则 r_1 和 r_2 的右侧相等，则两条决策规则 r_1 和 r_2 称为相等。

定义 3。决策规则系统 S 是一个有限的非空决策规则集合。

记 $A(S) = \bigcup_{r \in S} A(r)$、$n(S) = |A(S)|$, $D(S)$ 为 S 中决策规则的右侧集合，并记 $d(S)$ 为 S 中决策规则的最大长度。令 $n(S) > 0$。对于 $a_i \in A(S)$，令 $V_S(a_i) = \{\delta : a_i = \delta \in \bigcup_{r \in S} K(r)\}$ 且 $EV_S(a_i) = V_S(a_i) \cup \{*\}$，其中符号 $*$ 被解释为一个不属于集合 $V_S(a_i)$ 的数。记 $k(S) = \max\{|V_S(a_i)| : a_i \in A(S)\}$。若 $n(S) = 0$，则 $k(S) = 0$。我们用 Σ 表示决策规则系统的集合。

例 1. 考虑一个决策规则系统 $S = \{(a_1 = 0) \to 1, (a_1 = 1) \wedge (a_2 = 0) \to 2, (a_1 = 2) \wedge (a_3 = 0) \wedge (a_4 = 0) \to 3\}$。那么 $A(r_3) = \{a_1, a_3, a_4\}$, $K(r_3) = \{a_1 = 2, a_3 = 0, a_4 = 0\}$, 其中 r_3 表示来自 S 的第三条规则。$A(S) = \bigcup_{r \in S} A(r) = \{a_1, a_2, a_3, a_4\}$, $n(S) = |A(S)| = 4$, $D(S) = \{1, 2, 3\}$, $d(S) = 3$, $V_S(a_1) = \{\delta : a_1 = \delta \in \bigcup_{r \in S} K(r)\} = \{0, 1, 2\}$, $EV_S(a_1) = V_S(a_1) \cup \{*\} = \{0, 1, 2, *\}$ 且 $k(S) = \max\{|V_S(a_i)| : a_i \in A(S)\} = |V_S(a_1)| = 3$。

设 $S \in \Sigma$, $n(S) > 0$，且 $A(S) = \{a_{j_1}, \ldots, a_{j_n}\}$，其中 $j_1 < \cdots < j_n$。记 $V(S) = V_S(a_{j_1}) \times \cdots \times V_S(a_{j_n})$，以及 $EV(S) = EV_S(a_{j_1}) \times \cdots \times EV_S(a_{j_n})$。对于 $\delta = (\delta_1, \ldots, \delta_n) \in EV(S)$，记 $K(S, \delta) = \{a_{j_1} = \delta_1, \ldots, a_{j_n} = \delta_n\}$。

定义 4。我们称来自 S 的决策规则 r 对元组 $\delta \in EV(S)$ 是可实现的，如果 $K(r) \subseteq K(S, \delta)$。

显然，任何具有空左侧的规则都可由元组 δ 实现。

例 2。考虑决策规则系统 $S = \{r_1 : (a_1 = 0) \wedge (a_2 = 1) \to 1, r_2 : (a_1 = 1) \wedge (a_3 = 1) \to 2\}$ 以及元组 $\delta = (1, 1, 1) \in EV(S)$。那么，来自 S 的决策规则 r_2 可由元组 δ 实现，但 r_1 不可实现。

令 $V \in \{V(S), EV(S)\}$。现在我们定义三个与规则系统 S 相关的问题。

定义 5. 问题：对于有序对 (S, V)，所有规则 给定一个元组 $\delta \in V$，要求找出 S 中对于元组 δ 可实现的规则集合。

定义 6. 问题 所有决策 对于该对 (S, V)：对于给定的元组 $\delta \in V$，需要从 S 中找出一个满足以下条件的决策规则集合 Z：

- 来自 Z 的所有决策规则都可由元组 δ实现。

- 对于任意 $\sigma \in D(S) \setminus D(Z)$，来自 S 且右端等于 σ的任何决策规则，对于元组 δ 都是不可实现的。

定义7. 问题：某些规则关于对(S, V)：对于给定的元组 $\delta \in V$，需要从 S 中找出一组 Z 决策规则，使其满足以下条件：

- 来自 Z 的所有决策规则都可以通过元组 δ 实现。

- 如果 $Z = \emptyset$，那么 S 中的任何决策规则对于元组 δ都是不可实现的。

分别将针对成对对象 $(S, V(S))$ 和 $(S, EV(S))$ 的所有规则问题记为 $AR(S)$ 和 $EAR(S)$。分别将针对成对对象 $(S, V(S))$ 和 $(S, EV(S))$ 的所有决策问题记为 $AD(S)$ 和 $EAD(S)$。分别将针对成对对象 $(S, V(S))$ 和 $(S, EV(S))$ 的某些规则问题记为 $SR(S)$ 和 $ESR(S)$。

例3. 给定一个决策规则系统 $S = \{(a_1 = 0) \to 1, (a_1 = 0) \wedge (a_2 = 1) \to 1, (a_1 = 0) \wedge (a_2 = 1) \wedge (a_3 = 2) \to 2\}$ 以及一个元组 $\delta = (0, 1, 2) \in V(S)$。那么，$\{(a_1 = 0) \to 1, (a_1 = 0) \wedge (a_2 = 1) \to 1, (a_1 = 0) \wedge (a_2 = 1) \wedge (a_3 = 2) \to 2\}$ 是问题$AR(S)$的解，而元组 δ，$\{(a_1 = 0) \to 1, (a_1 = 0) \wedge (a_2 = 1) \wedge (a_3 = 2) \to 2\}$ 是问题 $AD(S)$的一个解，元组 δ则是问题 $\{(a_1 = 0) \to 1\}$ 的一个解；此外，$SR(S)$和元组 δ也是问题的一个解。

在特殊情况下，当 $n(S) = 0$时，S 中的所有规则都有一个空左侧。在这种情况下，自然可以考虑：(i) 将集合 S 视为问题 $AR(S)$ 和$EAR(S)$ 的解；(ii) 将集合 S 的任意子集 Z（满足 $D(Z) = D(S)$）视为问题$AD(S)$ 和 $EAD(S)$ 的解；(iii) 将集合 S 的任意非空子集 Z 视为问题 $SR(S)$ 和 $ESR(S)$ 的解。

设 $S \in \Sigma$，其中 Σ是决策规则系统的集合。我们用 $R_{SR}(S)$ 表示系统 S 的一个子系统，它由满足以下条件的所有规则 $r \in S$ 组成：不存在规则 $r' \in S$ 使得 $K(r') \subset K(r)$。

定义 8。如果 $R_{SR}(S) = S$，则系统 S 称为 SR-约简的 。

用 Σ_{SR} 表示 SR-约简决策规则系统的集合。

对于 $S \in \Sigma$，我们用 $R_{AD}(S)$ 表示系统 S 的一个子系统，该子系统由所有满足以下条件的规则 $r \in S$ 组成：不存在规则 $r' \in S$ 使得 $K(r') \subset K(r)$，且规则 r 和 r' 的右侧相同。

定义 9。系统 S 将被称为 AD-约简的 ，如果 $R_{AD}(S) = S$。

用 Σ_{AD} 表示由 AD 约简的决策规则系统集合。

示例4. 考虑一个决策规则系统 $S = \{(a_1 = 0) \wedge (a_2 = 1) \wedge (a_3 = 2) \to 1, (a_1 = 0) \wedge (a_2 = 1) \to 2, (a_1 = 0) \to 2\}$. 对于该系统，$R_{AD}(S) = \{(a_1 = 0) \wedge (a_2 = 1) \wedge (a_3 = 2) \to 1, (a_1 = 0) \to 2\}$ 和 $R_{SR}(S) = \{(a_1 = 0) \to 2\}$.

2.2 决策树

A 以根节点为根的有限有向树是指只有一个节点没有入边的有限有向树。这个节点称为根节点。没有出边的节点称为终端节点。不是终端节点的节点称为工作节点。以根节点为根的有限有向树中的完整路径是该树中由节点和边组成的序列 $\xi = v_1, d_1, \ldots, v_m, d_m, v_{m+1}$，其中 v_1 是根节点，v_{m+1} 是终端节点，并且对于 $i = 1, \ldots, m$，边 d_i 从节点 v_i 出发并进入节点 v_{i+1}。

我们将讨论两类决策树：o-决策树（普通决策树，简称 o-树）和 e-决策树（扩展决策树，简称 e-树）。

定义 10. 决策规则系统上的决策树 S 是一个带根节点 Γ 的有限有向标记树，满足以下条件：

- 树 Γ 的每个工作节点都标记为集合 $A(S)$中的一个属性。
- 设树 Γ的工作节点 v 标记为属性 a_i。若 Γ 是 o-树，则恰有 $|V_S(a_i)|$ 条边从节点 v 出发，且这些边分别标记为集合 $V_S(a_i)$中的两两不同元素。若 Γ是 e-树，则恰有 $|EV_S(a_i)|$ 条边从节点 v 出发，且这些边分别标记为集合 $EV_S(a_i)$中的两两不同元素。
- 树 Γ 的每个终端节点都标记为集合 S的一个子集。

设 Γ 是决策规则系统 S 上的一棵决策树。我们用 $CP(\Gamma)$ 表示树 Γ 中完全路径的集合。设 $\xi = v_1, d_1, \ldots, v_m, d_m, v_{m+1}$ 是 Γ 中的一条完全路径。我们为这条路径对应一个属性集合 $A(\xi)$ 和一个方程组 $K(\xi)$。如果 $m = 0$ 且 $\xi = v_1$，那么 $A(\xi) = \emptyset$ 且 $K(\xi) = \emptyset$。设 $m > 0$，并且对于 $j = 1, \ldots, m$，节点 v_j 标记为属性 a_{i_j}，边 d_j 标记为元素 $\delta_j \in \omega \cup \{*\}$。那么 $A(\xi) = \{a_{i_1}, \ldots, a_{i_m}\}$ 且 $K(\xi) = \{a_{i_1} = \delta_1, \ldots, a_{i_m} = \delta_m\}$。我们用 $\tau(\xi)$ 表示附属于节点 v_{m+1} 的决策规则集合。

例5. 考虑决策规则系统 $S = \{(a_1 = 0) \wedge (a_2 = 0) \rightarrow 1, (a_1 = 0) \wedge (a_2 = 1) \rightarrow 2, (a_1 = 1) \wedge (a_3 = 0) \rightarrow 3, (a_1 = 1) \rightarrow 4\}$。则该决策规则系统 S 上的 o-树 Γ_1 和 e-树 Γ_2 如图1所示，其中 r_1、r_2、r_3 和 r_4 分别是 S中的第一、第二、第三和最后一条决策规则。

设 ξ 是 o-树 Γ_1中的一条完整路径，该路径终止于标记有规则集合 $\{r_3, r_4\}$的终端节点。则 $A(\xi) = \{a_1, a_3\}$、$K(\xi) = \{a_1 = 1, a_3 = 0\}$ 和 $\tau(\xi) = \{r_3, r_4\}$。

定义 11. 方程组 $\{a_{i_1} = \delta_1, \ldots, a_{i_m} = \delta_m\}$，其中 a_{i_1}、.....、$a_{i_m} \in A$ 和δ_1、.....、$\delta_m \in \omega \cup \{*\}$，如果存在 l，$k \in \{1, \ldots, m\}$ 使得$l \neq k$、$i_l = i_k$和 $\delta_l \neq \delta_k$，则称其为不相容的 。如果该方程组不是不相容的，则称其为相容的。

设 S 为决策规则系统，Γ 为 S上的决策树。

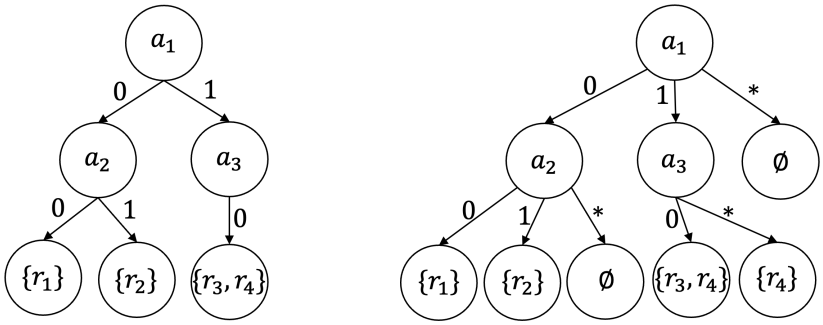


图1： 规则系统 S上的 o-决策树 Γ_1 和 e-决策树 Γ_2

定义 12. 如果 Γ解决 问题 $AR(S)$（分别为问题 $EAR(S)$），且 Γ 是一个 o-树（分别是一个 e-树），并且任何具有相容方程组 K（分别为 ξ）的路径 $\xi \in CP$（分别为 Γ）都满足以下条件，则称 Γ解决该问题：

- 对于任意决策规则 $r \in \tau(\xi)$，关系 $K(r) \subseteq K(\xi)$ 均成立。
- 对于任意决策规则 $r \in S \setminus \tau(\xi)$，方程组 $K(r) \cup K(\xi)$ 是不相容的。 tent.

示例 6. 设 S 是示例 4 中的决策规则系统。那么，图1中所示的决策树 Γ_1 和Γ_2 分别解决问题 $AR(S)$ 和 $EAR(S)$。

定义 13。 我们称 Γ解决 问题 $AD(S)$（分别为问题 $EAD(S)$），如果 Γ 是一棵 o-树（分别为一棵 e-树），并且任意路径 $\xi \in CP(\Gamma)$ 及其相容方程组 $K(\xi)$ 均满足以下条件：

- 对于任何决策规则 $r \in \tau(\xi)$，关系 $K(r) \subseteq K(\xi)$ 均成立。
- 如果 $r \in S \setminus \tau(\xi)$，且 r 的右侧不属于集合 $D(\tau(\xi))$，那么方程组 $K(r) \cup K(\xi)$ 不相容。

定义 14. 如果 Γ 求解 问题 $SR(S)$（分别为问题 $ESR(S)$），即 Γ 是一棵 o-树（分别为一棵 e-树），并且任何满足相容方程组 $K(\xi)$ 的路径 $\xi \in CP(\Gamma)$ 都满足以下条件：

- 对于任意决策规则 $r \in \tau(\xi)$，关系 $K(r) \subseteq K(\xi)$ 成立。
- 如果 $\tau(\xi) = \emptyset$，那么，对于任意决策规则 $r \in S$，方程组 $K(r) \cup K(\xi)$ 不相容。

例 7. 考虑决策规则系统 $S = \{(a_1 = 0) \wedge (a_2 = 0) \rightarrow 1$、$(a_1 =0) \wedge (a_2 = 1) \rightarrow 2$、$(a_1 = 1) \wedge (a_3 = 0) \rightarrow 3$、$(a_1 = 1) \rightarrow 3$、$(a_1 = 1) \rightarrow 4\}$ 以及图 2 所示的决策树Γ_1、 Γ_2和 Γ_3 。则决策树 Γ_1 解决问题 $AR(S)$， Γ_2解决 $AD(S)$，而 Γ_3 解决 $SR(S)$，其中 r_1、 r_2、 r_3、 r_4 和 r_5 分别是 S中的第一、第二、第三、第四和第五条决策规则。

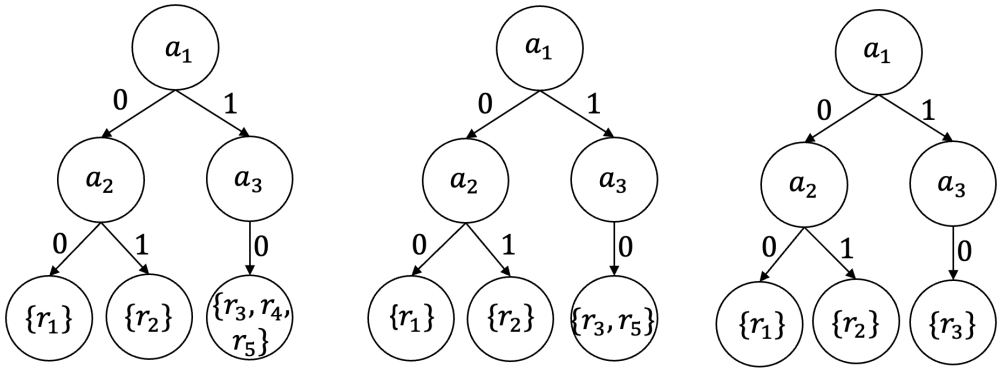


图 2： 决策树 Γ_1、 Γ_2 和 Γ_3

对于任意完整路径 $\xi \in CP(\Gamma)$，我们用 $h(\xi)$表示 ξ中的工作节点数。值 $h(\Gamma) = \max\{h(\xi) : \xi \in CP(\Gamma)\}$ 称为决策树 Γ 的 深度 。

设 S 为决策规则系统，且 $C \in \{AR, EAR, AD, EAD, SR, ESR\}$。我们用 $h_C(S)$ 表示 S 上解决问题 $C(S)$ 的决策树的最小深度。

设 $n(S) = 0$。如果 $C \in \{AR, EAR\}$，那么只有一棵解决问题$C(S)$的决策树。这棵树由一个标记为规则集 S 的节点组成。如果 $C \in \{AD, EAD\}$，那么解决问题 $C(S)$的决策树集合与这样的树的集合一致：每棵树都由一个标记为集合 S 的子集 Z 的节点组成，且 $D(Z) = D(S)$。如果$C \in \{SR, ESR\}$，那么解决问题 $C(S)$的决策树集合与这样的树的集合一致：每棵树都由一个标记为集合S的非空子集 Z 的节点组成。因此，如果 $n(S) = 0$，那么对于任意 $C \in \{AR, EAR, AD, EAD, SR, ESR\}$， $h_C(S) = 0$。

3 个辅助命题

在本节中，我们将首先给出 [12] 中的一些命题，然后证明一些新的命题。

设 S 是一个决策规则系统， Γ 是基于 S 的一棵 e-决策树。我们用 $o(\Gamma)$ 表示基于 S 的一棵 o- 树，它由树 Γ 删除所有满足以下条件的节点 v 得到：在 Γ 中，从根节点到节点 v 的路径包含一条标记为 $*$的边。与节点 v 一起，我们删除所有进入或离开 v 的边。

引理 1。（引理 *1 [12]*）设 S 是一个决策规则系统， $C \in \{AR, AD, SR\}$，且 Γ 是一棵基于 S 、解决问题 $EC(S)$ 的 *e*-决策树。那么，决策树 $o(\Gamma)$ 解决问题 $C(S)$。

引理 2。（引理 *2 [12]*）设 S 是一个决策规则系统， Γ 是一棵基于 S 的决策树。那么

（*a*）如果树 Γ 解决问题 $AR(S)$（分别为 $EAR(S)$），那么树 Γ 解决问题 $AD(S)$（分别为 $EAD(S)$）。

(b) 如果树 Γ 分别解决问题 $AD(S)$ ***(***$EAD(S)$，那么树 Γ 分别解决问题 $SR(S)$ ***(***$ESR(S)$。

引理 3. *(*引理 *3 [12])* 设 S 为一个决策规则系统。则以下不等式成立：

$$\begin{array}{ccccccc} h_{ESR}(S) & \leq & h_{EAD}(S) & \leq & h_{EAR}(S) & \leq & n(S) \\ \vee| & & \vee| & & \vee| & & \\ h_{SR}(S) & \leq & h_{AD}(S) & \leq & h_{AR}(S) & & \end{array}$$

设 S 是一个决策规则系统，$\alpha = \{a_{i_1} = \delta_1, \ldots, a_{i_m} = \delta_m\}$ 是一个相容方程组，使得 a_{i_1}，……，$a_{i_m} \in A$ 且 δ_1，……，$\delta_m \in \omega \cup \{*\}$。现在定义一个决策规则系统 S_α。设 r 是一条决策规则，且方程组 $K(r) \cup \alpha$ 相容。记 r_α 为从 r 中删除左侧所有属于 α 的方程后得到的决策规则。则 S_α 是满足 $r \in S$ 且方程组 $K(r) \cup \alpha$ 相容的决策规则 r_α 的集合。

引理 4。（引理 *6 [12]*）设 S 是一个决策规则系统，其中 $n(S) > 0$，$C \in \{EAR, AR, EAD, AD, ESR, SR\}$，$\alpha = \{a_{i_1} = \delta_1, \ldots, a_{i_m} = \delta_m\}$ 是一个相容方程组，使得 a_{i_1}，……，$a_{i_m} \in A(S)$，并且对于 $j = 1$，……，m，若 $C \in \{EAR, EAD, ESR\}$，则 $\delta_j \in EV_S(a_{i_j})$；若 $C \in \{AR, AD, SR\}$，则 $\delta_j \in V_S(a_{i_j})$。那么 $h_C(S) \geq h_C(S_\alpha)$。

我们将决策规则系统 S 对应为超图 $G(S)$，其节点集合为 $A(S)$，边集合为 $\{A(r) : r \in S\}$。超图 $G(S)$ 的一个节点覆盖是节点集合 $A(S)$ 的子集 B，满足：$A(r) \cap B \neq \emptyset$ 对于任意规则 $r \in S$，使得 $A(r) \neq \emptyset$。如果$A(S) = \emptyset$，则空集是超图 $G(S)$ 的唯一节点覆盖。记$\beta(S)$为超图 $G(S)$的节点覆盖的最小基数。

例 8。考虑决策规则系统 $S = \{(a_1 = 0) \wedge (a_2 = 1) \to 1, (a_1 =0) \wedge (a_3 = 1) \to 2, (a_4 = 0) \to 3\}$。可以证明，集合 $\{a_1, a_4\}$ 是超图 $G(S)$ 的节点覆盖，并且 $\beta(S) = 2$。

我们按如下方式定义系统 S 的子系统 $I_{SR}(S)$。如果 S 不包含长度为 0 的规则，那么 $I_{SR}(S) = S$。否则，$I_{SR}(S)$ 由 S 中所有长度为 0 的规则组成。

注记 1。注意，$I_{SR}(S) \neq S$ 当且仅当 S 同时包含一条长度为 0 的规则和一条长度大于 0 的规则时成立。

现在，我们定义系统 S 的子系统 $I_{AD}(S)$。用 $D_0(S)$ 表示来自 S 的决策规则的右侧集合，其中这些规则的长度等于 0。于是，子系统 $I_{AD}(S)$ 由 S 中所有长度为 0 的规则，以及 S 中右侧不属于 $D_0(S)$ 的所有规则组成。

注记 2。注意，$I_{AD}(S) \neq S$ 当且仅当 S 同时包含一条长度为 0 的规则和一条长度大于 0 且右侧相同的规则时成立。

例 9. 考虑决策规则系统 $S = \{(a_1 = 0) \to 1, (a_2 = 0) \to 2, \to 2\}$。对于这个系统，$I_{SR}(S) = \{\to 2\}$ 和 $I_{AD}(S) = \{(a_1 = 0) \to 1, \to 2\}$。

设 S 为一个决策规则系统。如果存在一个元组 $\delta \in V(S)$，使得对于任意决策规则 $r \in S$，方程组 $K(r) \cup K(S, \delta)$ 均不相容，则称该系统为不完备。否则，称系统 S 为完备。如果 $n(S) = 0$，则认为系统 S 完备。

引理5。 *(*引理 *7 [12])* 设 S 为一个决策规则系统。那么 *(a)* 若 $C \in \{EAR, AR\}$，则 $h_C(S) \geq \beta(S)$。*(b)* 若 $C \in \{AD, SR\}$，则 $h_{EC}(S) \geq \beta(I_C(S))$。*(c)* 若 $C \in \{AD, SR\}$ 且系统 S 不完备，则 $h_C(S) \geq \beta(S)$。

引理 6。 （引理 *8 [12]*）设 S 是一个决策规则系统。那么 *(a)* $h_{EAR}(S) \geq h_{AR}(S) \geq d(S)$。*(b)* 若 S 是一个 SR-约简系统，则 $h_{ESR}(S) \geq d(S)$。*(c)* 若 S 是一个 AD-约简系统，则 $h_{EAD}(S) \geq d(S)$。

引理 7。 （引理 *9 [12]*）设 S 是一个决策规则系统。那么 $h_{ESR}(S) = h_{ESR}(R_{SR}(S))$以及 $h_{EAD}(S) = h_{EAD}(R_{AD}(S))$。

设 $C \in \{AR, EAR, AD, EAD, SR, ESR\}$, S是一个决策规则系统， Γ是定义在 S上的一棵决策树。我们用 $L(\Gamma)$表示树 Γ 中的节点数，用 $T(\Gamma)$表示 Γ 中标记着两两不同决策规则集合的终端节点数。设 $L_C(S) = \min L(\Gamma)$，且 $T_C(S) = \min T(\Gamma)$，其中最小值取遍所有在 S上求解问题 $C(S)$的决策树 Γ。显然， $T_C(S) \leq L_C(S)$。

L引理 8. 设 S 是一个决策规则系统。则下列不等式成立 *:*

$$\begin{array}{ccccccccccc} L_{ESR}(S) & \leq & L_{EAD}(S) & \leq & L_{EAR}(S) & T_{ESR}(S) & \leq & T_{EAD}(S) & \leq & T_{EAR}(S) \\ \vee| & & \vee| & & \vee| & \vee| & & \vee| & & \vee| \\ L_{SR}(S) & \leq & L_{AD}(S) & \leq & L_{AR}(S), & T_{SR}(S) & \leq & T_{AD}(S) & \leq & T_{AR}(S). \end{array}$$

*P*证明。 显然，当 $n(S) = 0$ 时，所考虑的不等式成立。设 $n(S) > 0$.

设 Γ 是定义在 S 上的一棵 e-决策树。显然， $L(o(\Gamma)) \leq L(\Gamma)$ 且 $T(o(\Gamma)) \leq T(\Gamma)$。利用这些不等式和引理 1，我们得到 $L_{SR}(S) \leq L_{ESR}(S)$、 $L_{AD}(S) \leq L_{EAD}(S)$、 $L_{AR}(S) \leq L_{EAR}(S)$、 $T_{SR}(S) \leq T_{ESR}(S)$、 $T_{AD}(S) \leq T_{EAD}(S)$，以及 $T_{AR}(S) \leq T_{EAR}(S)$。

利用引理 2，我们得到 $L_{ESR}(S) \leq L_{EAD}(S) \leq L_{EAR}(S)$、 $L_{SR}(S) \leq L_{AD}(S) \leq L_{AR}(S)$、 $T_{ESR}(S) \leq T_{EAD}(S) \leq T_{EAR}(S)$，以及 $T_{SR}(S) \leq T_{AD}(S) \leq T_{AR}(S)$。 □

决策规则系统 S 可以表示为字母表上的一个字串

$$\{(,), a, =, \wedge, \rightarrow, 0, 1, ;\}$$

其中， ω 中的数字（属性索引、属性值以及决策规则的右侧）均采用二进制表示（即使用字母表 $\{0, 1\}$中的字串表示），符号";"用于分隔两条规则。该字串的长度称为决策规则系统 S 的大小，记作 $size(S)$。

定义15. 如果决策规则系统 S 满足以下条件，则称其为约简的：

- 如果 $n(s) = n$，那么 $A(S) \subseteq \{a_0, \ldots, a_n\}$。
- 如果 $|D(S)| = t$，那么 $D(S) \subseteq \{0, \ldots, t\}$。

- 如果 $k(S) = k$，那么对于任意 $a_i \in A(S)$， $V_S(a_i) \subseteq \{0, \ldots, k\}$。
- $d(S) \geq 1$。

设 S 是约简决策规则系统。显然， $n(S) \leq d(S)|S|$， $k(S) \leq |S|$，且 $|D(S)| \leq |S|$。因此，系统 S 中来自 ω 的最大数至多为 $d(S)|S|$。此类数的二进制表示长度至多为 $\log_2(d(S)|S|)+1$。 S 中每条规则的长度至多为 $d(S)$。可以证明，表示每条规则的字串长度（包括其后的符号“;”）至多为 $10d(S)(\log_2(d(S)|S|) + 1)$ 并且

$$size(S) \leq 10\,|S|\,d(S)(\log_2(d(S)\,|S|) + 1). \tag{1}$$

引理 9。令 n、 k、 $d \in \omega \setminus \{0\}$、 $d \geq 2$ 和 $k \geq 2$。则存在一个约化决策规则系统 S，使得 $n(S) = 2n + d$、 $k(S) = k$、 $d(S) = d$、 $|S| = 2n + k - 1$ 以及 $L_{SR}(S) \geq 2^n$。

证明。考虑一个决策规则系统 $S = S_1 \cup S_2 \cup S_3$，其中 $S_1 = \{(a_{2i-1} = 0) \wedge (a_{2i} = 0) \to 0, (a_{2i-1} = 1) \wedge (a_{2i} = 1) \to 0 : i = 1, \ldots, n\}$, $S_2 = \{(a_1 = 2) \wedge (a_2 = 2) \to 0, \ldots, (a_1 = k - 1) \wedge (a_2 = k - 1) \to 0\}$，且 $S_3 = \{(a_{2n+1} = 0) \wedge \cdots \wedge (a_{2n+d} = 0) \to 0\}$。若 $k = 2$，则 $S_2 = \emptyset$。显然，S 是约简的， $n(S) = 2n + d$, $k(S) = k$, $d(S) = d$，并且 $|S| = 2n + k - 1$。将满足 $\delta_{2i-1} + \delta_{2i} = 1$（对于 $i = 1, \ldots, n$）且 $\delta_{2n+1} = \cdots = \delta_{2n+d} = 1$ 的元组 $(\delta_1, \ldots, \delta_{2n+d}) \in \{0, 1\}^{2n+d}$ 的集合记为 Δ。显然， $|\Delta| = 2^n$，并且对于任意 $\delta \in \Delta$， S 中不存在一条可由元组 δ 实现的规则。

设 Γ 是关于 S 的决策树，它解决问题 $SR(S)$，并且满足 $L(\Gamma) = L_{SR}(S)$。设 $\delta = (\delta_1, \ldots, \delta_{2n+d}) \in \Delta$。显然， Γ 中存在一条完整路径 ξ，使得 $K(\xi) \subseteq K(S, \delta)$。显然，该路径的终端节点标记为空集。下面证明 $\{a_1 = \delta_1, \ldots, a_{2n} = \delta_{2n}\} \subseteq K(\xi)$。反设不然，则存在 $j \in \{1, \ldots, 2n\}$，使得方程 $a_j = \delta_j$ 不属于 $K(\xi)$。令 $j \in \{2i - 1, 2i\}$，其中 $i \in \{1, \ldots, n\}$。记规则 $(a_{2i-1} = 0) \wedge (a_{2i} = 0) \to 0$ 为 r_0，并记规则 $(a_{2i-1} = 1) \wedge (a_{2i} = 1) \to 0$ 为 r_1。显然， $r_0, r_1 \in S$，且方程组 $K(r_0) \cup K(\xi)$ 与 $K(r_1) \cup K(\xi)$ 中至少有一个相容，但这是不可能的。因此 $\{a_1 = \delta_1, \ldots, a_{2n} = \delta_{2n}\} \subseteq K(\xi)$。由此关系以及包含关系 $K(\xi) \subseteq K(S, \delta)$ 可知，在 Γ 中至少有 2^n 条两两不同的完整路径。因此 $L(\Gamma) \geq 2^n$，且 $L_{SR}(S) \geq 2^n$。

□

引理 10. 设 n、 $d \in \omega \setminus \{0\}$ 和 $d \geq 2$。那么存在一个约化的决策规则系统 S，使得 $n(S) = 2n + d$、 $k(S) = 1$、 $d(S) = d$、 $|S| = n + 1$，以及 $L_{ESR}(S) \geq 2^n$。

证明。考虑决策规则系统 $S = S_1 \cup S_2$，其中 $S_1 = \{(a_{2i-1} = 0) \wedge (a_{2i} = 0) \to 0 : i = 1, \ldots, n\}$ 以及 $S_2 = \{(a_{2n+1} = 0) \wedge \cdots \wedge (a_{2n+d} = 0) \to 0\}$。显然，$S$ 是约简的， $n(S) = 2n + d$, $d(S) = d$, $k(S) = 1$，并且 $|S| = n + 1$。将满足条件 $\{\delta_{2i-1}, \delta_{2i}\} = \{0, *\}$（对于 $i = 1, \ldots, n$）且 $\delta_{2n+1} = \cdots = \delta_{2n+d} = *$ 的元组 $(\delta_1, \ldots, \delta_{2n+d}) \in \{0, *\}^{2n+d}$ 组成的集合记为 Δ。显然， $|\Delta| = 2^n$，并且对于任意 $\delta \in \Delta$， S 中没有一条规则可由元组 δ 实现。设 Γ 是定义在 S 上、用于解决问题 $ESR(S)$ 的决策树，且满足 $L(\Gamma) = L_{ESR}(S)$。令 $\delta, \bar{\sigma} \in \Delta$ 和 $\delta \neq \bar{\sigma}$。显然， Γ 中存在两条完整路径 ξ, τ，使得 $K(\xi) \subseteq K(S, \delta)$ 且 $K(\tau) \subseteq K(S, \bar{\sigma})$。下面证明 $\xi \neq \tau$。反设： $\xi = \tau$。显然，路径 ξ 的终端节点

¯标记为空集。由于 $\delta \neq \bar{\sigma}$，存在 $i \in \{1,\ldots,n\}$，使得元组 δ 和 $\bar{\sigma}$ 的第 $(2i-1)$ 个数字与第 $(2i)$ 个数字不同。因此，属性 a_{2i-1} 和 a_{2i}未附着于路径 ξ上的任何节点。于是，方程组 $K(r) \cup K(\xi)$ 相容，其中 r 是来自 S 的规则 $(a_{2i-1}=0) \wedge (a_{2i}=0) \rightarrow 0$，但这是不可能的。因此 $\xi \neq \tau$。由此，树 Γ中至少有 2^n 条两两不同的完整路径。因此，我们有 $L(\Gamma) \geq 2^n$ 且 $L_{ESR}(S) \geq 2^n$。 □

引理 11。 ***(a)*** 设 n、$d \in \omega \setminus \{0\}$。则存在一个约化决策规则系统 S，使得 $n(S)=n+d$、$k(S)=1$、$d(S)=d$、$|S|=n+1$，且 $T_{EAD}(S) \geq 2^{n+1}$。

(b) 设 n、d、$k \in \omega \setminus \{0\}$ 和 $k \geq 2$。则存在一个约化决策规则系统 S，使得 $n(S)=n+d$、$k(S)=k$、$d(S)=d$、$|S|=2n+k-1$，且 $T_{AD}(S) \geq 2^n$。

证明。 (a) 考虑一个决策规则系统 $S = \{(a_1=0) \rightarrow 1, (a_2=0) \rightarrow 2, \ldots, (a_n=0) \rightarrow n, (a_{n+1}=0) \wedge \cdots \wedge (a_{n+d}=0) \rightarrow n+1\}$。显然，$S$ 是约简的，$n(S)=n+d$、$k(S)=1$、$d(S)=d$、$|S|=n+1$，且问题 $EAD(S)$ 有 2^{n+1} 个两两不同的解。因此，任何解决问题 $EAD(S)$ 的决策树都至少有 2^{n+1} 个终端节点，这些终端节点分别标记着两两不同的决策规则集合。因此$T_{EAD}(S) \geq 2^{n+1}$。

(b) 考虑决策规则系统 $S = S_1 \cup S_2 \cup S_3$，其中 $S_1 = \{(a_i=0) \rightarrow 2i-1, (a_i=1) \rightarrow 2i\colon\ i=1,\ldots,n\}$, $S_2 = \{(a_{n+1}=0) \wedge \cdots \wedge (a_{n+d}=0) \rightarrow 0\}$, and$S_3 = \{(a_1=2) \rightarrow 0, \ldots, (a_1=k-1) \rightarrow 0\}$。若 $k=2$，则 $S_3 = 0$。显然，S 是约简的，$n(S) = n+d$, $k(S) = k$, $d(S) = d$，且 $|S| = 2n+k-1$。显然，问题 $AD(S)$ 至少有 2^n 个两两不同的解。因此，任何求解问题 $AD(S)$ 的决策树都至少有 2^n 个终端节点，它们分别标记着两两不同的决策规则集合。因此 $T_{AD}(S) \geq 2^n$。 □

令 $k, d \in \omega \setminus \{0\}$。记 $\Sigma(k,d) = \{S \in \Sigma\colon\ k(S)=k, d(S)=d\}$。我们将仅考虑用于构造决策树的顺序算法，并根据算法输入中决策规则系统的大小评估其时间复杂度。

引理 12。 令 $d \in \omega \setminus \{0\}$ 和 $C \in \{SR, AD, AR\}$。则存在一个多项式时间算法，对于给定的决策规则系统 $S \in \Sigma(1,d)$，构造一棵解决问题 $C(S)$的决策树。

证明。 令 $S \in \Sigma(1,d)$，$A(S) = \{a_{i_1}, \ldots, a_{i_m}\}$、$i_1 < \cdots < i_m$以及 $V(S) = \{(\delta_1, \ldots, \delta_m)\}$。我们构造一棵决策树 Γ，它由一条完整路径 $\xi = v_1, d_1, \ldots, v_m, d_m, v_{m+1}$组成，使得对于 $j=1,\ldots,m$，节点 v_j 标记属性 a_{i_j}，边 d_j 标记数字 δ_j，节点 v_{m+1} 标记决策规则集合 S。显然，Γ 解决了问题 $C(S)$，且所考虑算法具有多项式时间复杂度。

□

引理 13. 设 $k \in \omega \setminus \{0\}$ 且 $C \in \{SR, ESR\}$。那么存在一种多项式时间算法，对于给定的决策规则系统 $S \in \Sigma(k,1)$，构造一棵解决问题 $C(S)$的决策树。

证明。设 $S \in \Sigma(k,1)$，且 S 包含一条长度为 0 的规则 r。构造一棵仅包含一个节点 Γ_0的决策树，该节点标记为集合 $\{r\}$。显然， Γ_0 解决问题$C(S)$，且所考虑算法的时间复杂度为多项式级。

现在假设 S不包含长度为 0 的规则，并且 $A(S) = \{a_{i_1}, \ldots, a_{i_n}\}$。令 $j \in \{1, \ldots, n\}$ 和 $\delta \in V_S(a_{ij})$。显然，在系统 S中存在一条决策规则，其左侧等于 $a_{i_j} = \delta$。将该规则记为 $r(j, \delta)$。

让我们构造一个决策树 Γ_1。令 $V_S(a_{i_1}) = \{\delta_1, \ldots, \delta_m\}$。该决策树 Γ_1由节点 $v_0, v_1, \ldots, v_m$ 以及边 $d_1, \ldots, d_m$ 组成。节点 v_0 标记为属性 a_{i_1}，节点 v_j 标记为集合 $\{r(1, \delta_j)\}$，$j = 1, \ldots, m$。对于$j = 1, \ldots, m$，边 d_j标记为数字 δ_j，边 d_j从节点 v_0出发并进入节点 v_j。显然， Γ_1 解决了问题 $SR(S)$，且所考虑的算法具有多项式时间复杂度。

让我们构造一棵决策树 Γ_2。树 Γ_2包含一条完整路径 $\xi = v_1, d_1, \ldots, v_n, d_n, v_{n+1}$。对于 $j = 1, \ldots, n$，节点v_j标记为属性a_{i_j}，边 d_j标记为符号 $*$。节点 v_{n+1}标记为空集。设 $j \in \{1, \ldots, n\}$和 $V_S(a_{i_j}) = \{\delta_1, \ldots, \delta_m\}$。除边 d_j外，边 $d_{j1}, \ldots, d_{jm}$也从节点v_j引出。这些边分别标记为数字 $\delta_1, \ldots, \delta_m$。边 $d_{j1}, \ldots, d_{jm}$分别进入节点$v_{j1}, \ldots, v_{jm}$，这些节点标记为集合 $\{r(j, \delta_1)\}, \ldots, \{r(j, \delta_m)\}$。树 Γ 不包含任何其他节点和边。显然， Γ_2 解决了问题 $ESR(S)$，且所考虑算法的时间复杂度为多项式。

□

4 决策树的构造

如上所述，我们只考虑用于构造决策树的顺序算法，并根据算法输入中决策规则系统的大小来评估其时间复杂度。我们假设算法在每个单位时间内至多向正在构造的决策树添加一个节点。

令 $C \in \{SR, ESR, AD, EAD, AR, EAR\}$。用 $P_{DT}(C)$ 表示满足以下条件的配对 $(k, d) \in (\omega \setminus \{0\})^2$ 的集合：存在一个多项式时间算法，对于任意决策规则系统 $S \in \Sigma(k, d)$，都能构造一棵解决问题$C(S)$ 的决策树。

定理 1. *(a)* $P_{DT}(ESR) = \{(k, 1) : k \in \omega \setminus \{0\}\}$, *(b)* $P_{DT}(SR) = \{(k, 1), (1, d) : k, d \in \omega \setminus \{0\}\}$, *(c)* $P_{DT}(EAD) = P_{DT}(EAR) = \emptyset$, 以及 *(d)* $P_{DT}(AD) = P_{DT}(AR) = \{(1, d) : d \in \omega \setminus \{0\}\}$。

证明。令 $C \in \{SR, ESR, AD, EAD, AR, EAR\}$，$(k, d) \in (\omega \setminus \{0\})^2$ 并假设存在一个多项式时间算法，该算法对于给定的决策规则系统 $S \in \Sigma(k, d)$ 构造一个解决问题 $C(S)$ 的决策树。则存在一个多项式 π，使得对于任意 $S \in \Sigma(k, d)$，$L_C(S) \leq \pi(size(S))$。如果 S是约简的，那么根据 (1)，$size(S) \leq 10\,|S|\,d(S)(\log_2(d(S)\,|S|) + 1)$。我们知道，对于系统 $S \in \Sigma(k, d)$，$d(S) \leq d$。因此存在一个多项式 ϱ，使得对于任意约简系统 $S \in \Sigma(k, d)$，$L_C(S) \leq \varrho(|S|)$。

(a) 利用引理 13，我们得到 $\{(k, 1) : k \in \omega \setminus \{0\}\} \subseteq P_{DT}(ESR)$。令 $k, d \in \omega \setminus \{0\}$和 $d > 1$。现在我们证明 $(k, d) \notin P_{DT}(ESR)$。假设相反的情况成立：$(k, d) \in$

$P_{DT}(ESR)$。那么存在一个多项式 ϱ，使得对于任意约简系统 $S \in \Sigma(k,d)$，都有 $L_{ESR}(S) \leq \varrho(|S|)$。令 $k=1$。利用引理 10，我们得到，对于任意 $n \in \omega \setminus \{0\}$，都存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = n+1$ 且 $L_{ESR}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有 $2^n \leq \varrho(n+1)$，但这是不可能的。令 $k > 1$。利用引理 9，我们得到，对于任意 $n \in \omega \setminus \{0\}$，都存在一个约化决策规则系统$S \in \Sigma(k,d)$，使得 $|S| = 2n+k-1$ 且 $L_{SR}(S) \geq 2^n$。根据引理 8，$L_{ESR}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有 $2^n \leq \varrho(2n+k-1)$，但这是不可能的，因为对于所考虑的决策规则系统，k 是一个常数。因此，如果 $d > 1$，则 $(k,d) \notin P_{DT}(ESR)$。于是，$P_{DT}(ESR) = \{(k,1) : k \in \omega \setminus \{0\}\}$。

(b) 利用引理12和引理13，我们得到 $\{(k,1),(1,d) : k,d \in \omega \setminus \{0\}\} \subseteq P_{DT}(SR)$。令 $k, d \in \omega \setminus \{0,1\}$。现在证明 $(k,d) \notin P_{DT}(SR)$。反设：$(k,d) \in P_{DT}(SR)$。那么存在一个多项式 ϱ，使得对于任意约简系统 $S \in \Sigma(k,d)$，都有 $L_{SR}(S) \leq \varrho(|S|)$。利用引理 9，我们得到，对于任意 $n \in \omega \setminus \{0\}$，都存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = 2n+k-1$ 且 $L_{SR}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有 $2^n \leq \varrho(2n+k-1)$；但这是不可能的，因为对于所考虑的决策规则系统，k 是常数。因此，当 $k > 1$ 且 $d > 1$时，$(k,d) \notin P_{DT}(SR)$。于是，$P_{DT}(SR) = \{(k,1),(1,d) : k,d \in \omega \setminus \{0\}\}$。

(c) 令 $k,d \in \omega \setminus \{0\}$。现在证明 $(k,d) \notin P_{DT}(EAD)$。反设：$(k,d) \in P_{DT}(EAD)$。于是存在一个多项式 ϱ，使得对于任意约简系统 $S \in \Sigma(k,d)$，$L_{EAD}(S) \leq \varrho(|S|)$。令 $k=1$。根据引理 11，我们得到，对于任意$n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = n+1$ 且$T_{EAD}(S) \geq 2^{n+1}$。显然，$L_{EAD}(S) \geq T_{EAD}(S)$，且 $L_{EAD}(S) \geq 2^{n+1}$。因此，对于任意 $n \in \omega \setminus \{0\}$，$2^{n+1} \leq \varrho(n+1)$，但这是不可能的。令 $k > 1$。根据引理 11，我们得到，对于任意 $n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = 2n+k-1$ 且 $T_{AD}(S) \geq 2^n$。根据引理 8，$T_{EAD}(S) \geq T_{AD}(S)$。显然，$L_{EAD}(S) \geq T_{EAD}(S)$，且 $L_{EAD}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$，$2^n \leq \varrho(2n+k-1)$，但这是不可能的，因为对于所考虑的决策规则系统，$k$ 是一个常数。因此 $(k,d) \notin P_{DT}(EAD)$。从而，$P_{DT}(EAD) = \emptyset$。

设 $k,d \in \omega \setminus \{0\}$。现在证明 $(k,d) \notin P_{DT}(EAR)$。反设：$(k,d) \in P_{DT}(EAR)$。于是存在一个多项式 ϱ，使得对于任意约简系统 $S \in \Sigma(k,d)$，都有 $L_{EAR}(S) \leq \varrho(|S|)$。设 $k=1$。由引理 11 可得，对于任意$n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = n+1$ 且$T_{EAD}(S) \geq 2^{n+1}$。显然，$L_{EAD}(S) \geq T_{EAD}(S)$。由引理 8，$L_{EAR}(S) \geq L_{EAD}(S)$，且 $L_{EAR}(S) \geq 2^{n+1}$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有 $2^{n+1} \leq \varrho(n+1)$，但这是不可能的。设 $k > 1$。由引理 11 可得，对于任意 $n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = 2n+k-1$ 且 $T_{AD}(S) \geq 2^n$。由引理 8，$T_{EAR}(S) \geq T_{AD}(S)$。显然，$L_{EAR}(S) \geq T_{EAR}(S)$，且 $L_{EAR}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有$2^n \leq \varrho(2n+k-1)$，但这是不可能的，因为对于所考虑的决策规则系统，$k$ 是一个常数。因此 $(k,d) \notin P_{DT}(EAR)$。于是，$P_{DT}(EAR) = \emptyset$。

（d）利用引理 12，我们得到 $\{(1,d) : d \in \omega \setminus \{0\}\} \subseteq P_{DT}(AD)$ 和 $\{(1,d) : d \in \omega \setminus \{0\}\} \subseteq P_{DT}(AR)$。

令 $k,d \in \omega \setminus \{0\}$ 和 $k \geq 2$。现在我们证明 $(k,d) \notin P_{DT}(AD)$。反设 $(k,d) \in P_{DT}(AD)$。那么存在一个多项式 ϱ，使得对于任意约简系统 $S \in \Sigma(k,d)$，都有 $L_{AD}(S) \leq \varrho(|S|)$。利用引理 11，我们得到，对于任意 $n \in \omega \setminus \{0\}$，有

存在一个约化决策规则系统 $S \in \Sigma(k,\ d)$，使得 $|S| = 2n+k-1$并且 $T_{AD}(S) \geq 2^n$。显然， $L_{AD}(S) \geq T_{AD}(S)$，且 $L_{AD}(S) \geq 2^n$。因此，对于任意$n \in \omega \setminus \{0\}$，都有 $2^n \leq \varrho(2n+k-1)$，但这是不可能的，因为 k 对于所考虑的决策规则系统是一个常量。因此，$(k,\ d) \notin P_{DT}(AD)$。于是， $P_{DT}(AD) = \{(1,\ d)\colon d \in \omega \setminus \{0\}\}$。

设 k、$d \in \omega \setminus \{0\}$ 和 $k \geq 2$。现在证明 $(k,d) \notin P_{DT}(AR)$。反设：$(k,d) \in P_{DT}(AR)$。于是存在一个多项式 ϱ，使得对于任意约简系统 $S \in \Sigma(k,d)$，都有 $L_{AR}(S) \leq \varrho(|S|)$。利用引理 11，我们得到，对于任意 $n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = 2n+k-1$ 且 $T_{AD}(S) \geq 2^n$。显然， $L_{AD}(S) \geq T_{AD}(S)$。根据引理 8， $L_{AR}(S) \geq L_{AD}(S)$，且 $L_{AR}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有 $2^n \leq \varrho(2n+k-1)$，但这是不可能的，因为对于所考虑的决策规则系统，k 是常数。因此，$(k,d) \notin P_{DT}(AR)$。于是，$P_{DT}(AR) = \{(1,d)\colon\ d \in \omega \setminus \{0\}\}$。 □

5 无环决策图的构造

除了决策树之外，我们还将考虑 无环决策图 ，其定义与决策树类似，但这里考虑的是带根节点且没有有向环的有限有向图，而不是有限有向树。注意，任何决策树都是无环决策图。设 $C \in \{SR, ESR, AD, EAD, AR, EAR\}$。对于决策规则系统 S，我们用 $L_C^{DG}(S)$ 表示能解决问题 $C(S)$ 的无环决策图中的最少节点数。可以证明， $L_C^{DG}(S) \geq T_C(S)$。

首先，我们描述一种将在本节和下一节中使用的构造。设 $C \in \{SR, ESR, AD, EAD, AR, EAR\}$、 $S \in \Sigma$ 和 $r \in S$ 是形式为 $(a_{i_1} = \delta_1) \wedge \dots \wedge (a_{i_m} = \delta_m) \to \sigma$ 的决策规则，其中 $m > 0$。对于 $j = 1,\ \dots,\ m$，若 $C \in \{SR, AD, AR\}$，则令 $V_S^C(a_{i_j}) = V_S(a_{i_j})$；若 $C \in \{ESR, EAD, EAR\}$，则令 $V_S^C(a_{i_j}) = EV_S(a_{i_j})$。现在我们来描述一个无环决策图 $G_S^C(r)$。图 $G_S^C(r)$ 包含 $m+2$ 个节点 $v_1,\ \dots,\ v_m$、v_{m+1}、v_{m+2}。节点 v_1 是 $G_S^C(r)$ 的根节点。对于 $j = 1,\ \dots,\ m$，节点 v_j 标记为属性 a_{i_j}，节点 v_{m+1} 标记为集合 $\{r\}$，节点 v_{m+2} 标记为空集。对于 $j = 1,\ \dots,\ m$，恰有 $\left|V_S^C(a_{i_j})\right|$ 条边从节点 v_j 引出。这些边分别标记为集合 $V_S^C(a_{i_j})$ 中互不相同的元素。标记为数字 δ_j 的边指向节点 v_{j+1}。所有其他边均指向节点 v_{m+2}。图 $G_S^C(r)$ 不包含其他节点和边。

我们仅考虑用于构造无环决策图的顺序算法，并根据算法输入中决策规则系统的大小来评估其时间复杂度。我们假设，在每个单位时间内，算法至多可以向正在构造的无环决策图中添加一个节点。

设 $C \in \{SR, ESR, AD, EAD, AR, EAR\}$。用 $P_{DG}(C)$ 表示满足以下条件的 $(k,\ d) \in (\omega \setminus \{0\})^2$ 对的集合：存在一个多项式时间算法，对于任意决策规则系统 $S \in \Sigma(k,\ d)$，都能构造一个解决问题 $C(S)$ 的无环决策图。

定理 2. *(a)* $P_{DG}(SR) = P_{DG}(ESR) = (\omega \setminus \{0\})^2$, *(b)* $P_{DG}(AD) = P_{DG}(AR) = \{(1,d)\colon d \in \omega \setminus \{0\}\}$, 以及 *(c)* $P_{DG}(EAD) = P_{DG}(EAR) = \emptyset$。

证明。设 $C \in \{SR, ESR, AD, EAD, AR, EAR\}$ 和 $(k,d) \in (\omega \setminus \{0\})^2$。假设存在一种多项式时间算法，能够针对给定的决策规则系统 $S \in \Sigma(k,d)$，构造一个解决问题 $C(S)$ 的无环决策图。那么存在一个多项式π ，使得对于任意 $S \in \Sigma(k,d)$，$L_C^{DG}(S) \leq \pi(size(S))$。如果 S 是约简的，那么根据 (1)，$size(S) \leq 10\,|S|\,d(S)(\log_2(d(S)\ |S|) + 1)$。我们知道，对于系统 $S \in \Sigma(k,d)$，$d(S) \leq d$。因此，存在一个多项式 ϱ ，使得对于任意约简系统 $S \in \Sigma(k,d)$，$T_C(S) \leq L_C^{DG}(S) \leq \varrho(|S|)$。

(a) 令 $C \in \{SR, ESR\}$。现在我们证明 $P_{DG}(C) = (\omega \setminus \{0\})^2$。此外，我们证明存在一个多项式时间算法，对于给定的决策规则系统 $S \in \Sigma$，可以构造一个用于解决问题 $C(S)$ 的无环决策图。

令 $S \in \Sigma$。如果系统 S 包含一条长度为 0 的决策规则 r ，那么由一个标记为空集 $\{r\}$ 的节点组成的无环决策图可以解决问题 $C(S)$。设系统 S 不包含长度为 0 的规则，且 $S = \{r_1, \ldots, r_t\}$。我们将图 $G_S^C(r_1) \ldots, G_S^C(r_t)$连接起来（决策规则 $r \in S$ 对应的图 $G_S^C(r)$的描述见第 5 节开头）。为此，对于 $j = 1, \ldots, t-1$，用图 $G_S^C(r_{j+1})$的根节点替换图 $G_S^C(r_j)$ 中标记为空集的节点。将所得图记为G_S^C。可以证明，图 G_S^C 解决了问题 $C(S)$。显然，所考虑的算法具有多项式时间复杂性。因此， $P_{DG}(C) = (\omega \setminus \{0\})^2$。

(b) 根据引理 12，我们得到 $\{(1,\ d)\colon\ d \in \omega \setminus \{0\}\} \subseteq P_{DG}(AD)$ 以及 $\{(1,\ d)\colon\ d \in \omega \setminus \{0\}\} \subseteq P_{DG}(AR)$。

设 k、$d \in \omega \setminus \{0\}$ 和 $k \geq 2$。现在我们证明 $(k,d) \notin P_{DG}(AD)$。反设：$(k,d) \in P_{DG}(AD)$。那么存在一个多项式 ϱ ，使得对于任意约简系统 $S \in \Sigma(k,d)$，$T_{AD}(S) \leq \varrho(|S|)$。根据引理 11，我们得到，对于任意 $n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = 2n + k - 1$ 且 $T_{AD}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$， $2^n \leq \varrho(2n + k - 1)$，但这是不可能的，因为对于所考虑的决策规则系统，$k$ 是一个常数。因此 $(k,d) \notin P_{DG}(AD)$。于是，$P_{DG}(AD) = \{(1,d)\colon\ d \in \omega \setminus \{0\}\}$。

令 $k, d \in \omega \setminus \{0\}$ 和 $k \geq 2$。现在我们证明 $(k,d) \notin P_{DG}(AR)$。反设：$(k,d) \in P_{DG}(AR)$。那么存在一个多项式 ϱ ，使得对于任意约简系统 $S \in \Sigma(k,d)$，都有 $T_{AR}(S) \leq \varrho(|S|)$。由引理 11 可知，对于任意 $n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = 2n+k-1$ 且 $T_{AD}(S) \geq 2^n$。由引理 8， $T_{AR}(S) \geq T_{AD}(S)$，且 $T_{AR}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$ ，都有 $2^n \leq \varrho(2n + k - 1)$；但这是不可能的，因为对于所考虑的决策规则系统， k 是常数。因此，$(k,d) \notin P_{DT}(AR)$。于是， $P_{DG}(AR) = \{(1,d)\colon\ d \in \omega \setminus \{0\}\}$。

(c) 令 $k, d \in \omega \setminus \{0\}$。现在证明 $(k,d) \notin P_{DG}(EAD)$。反设不然：$(k,d) \in P_{DG}(EAD)$。那么存在一个多项式 ϱ ，使得对于任意约简系统 $S \in \Sigma(k,d)$，都有 $T_{EAD}(S) \leq \varrho(|S|)$。令 $k = 1$。利用引理 11，我们得到：对于任意$n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = n + 1$ 且 $T_{EAD}(S) \geq 2^{n+1}$。因此，对于任意 $n \in \omega \setminus \{0\}$ ，都有 $2^{n+1} \leq \varrho(n + 1)$，但这是不可能的。令 $k > 1$。利用引理 11，我们得到：对于任意 $n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k,d)$，使得 $|S| = 2n + k - 1$ 且 $T_{AD}(S) \geq 2^n$。由引理 8，$T_{EAD}(S) \geq T_{AD}(S)$ 且 $T_{EAD}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有 $2^n \leq \varrho(2n + k - 1)$，但这是不可能的，因为对于所考虑的决策规则系统，$k$ 是一个常数。因此 $(k,d) \notin P_{DG}(EAD)$。于是， $P_{DG}(EAD) = \emptyset$。

令 $k, d \in \omega \setminus \{0\}$。现在证明 $(k, d) \notin P_{DG}(EAR)$。反设不然：$(k, d) \in P_{DG}(EAR)$。那么存在一个多项式 ϱ，使得对于任意约简系统 $S \in \Sigma(k, d)$，都有 $T_{EAR}(S) \leq \varrho(|S|)$。令 $k = 1$。利用引理 11，我们得到：对于任意$n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k, d)$，使得 $|S| = n + 1$且 $T_{EAD}(S) \geq 2^{n+1}$。由引理 8，$T_{EAR}(S) \geq T_{EAD}(S)$ 且 $T_{EAR}(S) \geq 2^{n+1}$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有$2^{n+1} \leq \varrho(n + 1)$，但这是不可能的。令 $k > 1$。利用引理 11，我们得到：对于任意 $n \in \omega \setminus \{0\}$，存在一个约化决策规则系统 $S \in \Sigma(k, d)$，使得 $|S| = 2n + k - 1$ 且 $T_{AD}(S) \geq 2^n$。由引理 8，$T_{EAR}(S) \geq T_{AD}(S)$ 且$T_{EAR}(S) \geq 2^n$。因此，对于任意 $n \in \omega \setminus \{0\}$，都有 $2^n \leq \varrho(2n + k - 1)$，但这是不可能的，因为对于所考虑的决策规则系统，$k$ 是一个常数。因此 $(k, d) \notin P_{DG}(EAR)$。于是，$P_{DG}(EAR) = \emptyset$。

□

6 带写入的无环决策图的构造

通过考虑带写入的无环决策图，可以避免与某个问题存在大量两两不同的解相关的困难。这类图除了具有标记了属性的工作节点外，还具有写入节点和一个终端节点。图中的一个节点被指定为根节点。每个写入节点都标记有一条决策规则，且只有一条出边。此边未标记。唯一的终端节点标记有字母 W，表示一个决策规则集合 W。在带写入的无环决策图工作期间，这个集合可以改变。在工作开始时（即我们位于图的根节点时），$W = \emptyset$。如果在工作期间到达一个标记有决策规则的写入节点，我们就将这条规则添加到集合 W。当到达终端节点时，此时形成的集合 W 就是所考虑的带写入的无环决策图的工作结果。

我们仅考虑用于构造带写入的无环决策图的顺序算法，并根据算法输入中决策规则系统的大小来评估其时间复杂度。我们假设，在每个单位时间内，算法最多只能向正在构造的带写入的无环决策图中添加一个节点。

设 $C \in \{SR, ESR, AD, EAD, AR, EAR\}$。用 $P_{DGW}(C)$ 表示满足以下条件的有序对集合：存在一个多项式时间算法，对于任意决策规则系统 $S \in \Sigma(k, d)$，构造一个带写入的无环决策图，以解决问题 $C(S)$。

定理 3。 $P_{DGW}(SR) = P_{DGW}(AD) = P_{DGW}(AR) = P_{DGW}(ESR) = P_{DGW}(EAD) = P_{DGW}(EAR) = (\omega \setminus \{0\})^2$。

证明。设 $C \in \{SR, ESR, AD, EAD, AR, EAR\}$、$S \in \Sigma$ 和 $r \in S$。现在我们描述一个写入为 $D_S^C(r)$ 的无环决策图。设决策规则 r 的长度等于 0。于是，$D_S^C(r)$ 包含两个节点 v_1 和 v_2，以及一条从节点 v_1 离开并进入节点 v_2的边。节点 v_1 是 $D_S^C(r)$ 的根节点。节点 v_1 标记为决策规则 r，节点 v_2 标记为字母 W。设 r 是形式为 $(a_{i_1} = \delta_1) \wedge \cdots \wedge (a_{i_m} = \delta_m) \rightarrow \sigma$ 的决策规则，其中 $m > 0$。那么，图 $D_S^C(r)$ 由图 $G_S^C(r)$（该图在第 5 节开头定义）按以下方式构造。

我们不使用集合 $\{r\}$，而是用规则 r标记节点 v_{m+1}。我们不使用空集，而是用字母 W标记节点 v_{m+2}。我们添加一条从节点 v_{m+1} 离开并进入节点 v_{m+2}的边。

现在我们证明 $P_{DGW}(C) = (\omega \setminus \{0\})^2$。事实上，我们证明存在一个多项式时间算法，对于给定的决策规则系统 $S \in \Sigma$，该算法构造一个带写入的无环决策图 G，以解决问题 $C(S)$。设 $S = \{r_1, \ldots, r_t\}$。首先，我们构造图 $D_S^C(r_1) \ldots, D_S^C(r_t)$。然后将这些图连接起来。为此，对于 $j = 1, \ldots, t-1$，将图 $D_S^C(r_j)$ 中标记为字母 W 的节点替换为图$D_S^C(r_{j+1})$的根节点。将得到的带写入的无环决策图记为 G。可以证明，图 G 解决了问题 $C(S)$。显然，所考虑算法的时间复杂度为多项式。

□

鉴于构造决策树时遇到的困难，可以转而研究无环决策图和带写入的无环决策图。然而，在构造这些图时，应同时尽可能减小深度，同时避免节点数量过度增加。此类双准则优化的可能性将作为未来一项专门研究的课题。在下一节中，我们将考虑另一种方法，该方法基于模拟决策树对给定属性值元组的处理过程。

7 基于节点覆盖的界与算法

本节将继续探讨决策树，但我们不再构造完整的决策树，而是仅考虑多项式时间算法：对于给定的属性值元组，这些算法描述决策树在该元组上的运行过程。为此，我们将研究决策树深度的界，以及基于与所研究的决策规则系统相对应的超图的节点覆盖、用于描述决策树运行过程的算法。

7.1 界

本节将研究用于解决问题的决策树最小深度的界。这些界取决于规则的最大长度，以及基于与规则系统相对应的超图的节点覆盖的参数。下一节中，我们将考虑多项式时间算法，这些算法基于本节提出的思想，对决策树在给定属性值元组上的运行过程进行建模。

设 S 是具有 $n(S) > 0$ 的决策规则系统。我们用 S^+ 表示 S 的子系统，其中仅包含长度为 $d(S)$ 的规则。记 $\beta^+(S) = \beta(S^+)$，其中 $\beta(S^+)$ 是超图 $G(S^+)$ 的节点覆盖的最小基数。显然，$\beta(S^+) \leq \beta(S)$。设 $C \in \{EAR, EAD, ESR\}$。我们用 $E_C(S)$ 表示相容方程组 $\{a_{i_1} = \delta_1, \ldots, a_{i_m} = \delta_m\}$ 的集合，使得 $a_{i_j} \in A(S)$ 且对于 $j = 1, \ldots, m$，有 $\delta_j \in EV_S(a_{i_j})$。设$C \in \{AR, AD, SR\}$。我们用 $E_C(S)$ 表示相容方程组 $\{a_{i_1} = \delta_1, \ldots, a_{i_m} = \delta_m\}$ 的集合，使得 $a_{i_j} \in A(S)$ 且对于 $j = 1, \ldots, m$，有 $\delta_j \in V_S(a_{i_j})$。记 $I_{EAD}(S) = I_{AD}(S)$，且 $I_{ESR}(S) = I_{SR}(S)$。

设 $C \in \{AR, AD, SR, EAR\}$。记为

$$\begin{aligned} \beta_C(S) &= \max\{\beta(S_\alpha) : \alpha \in E_C(S)\}, \\ \beta_C^+(S) &= \max\{\beta^+(S_\alpha) : \alpha \in E_C(S)\}. \end{aligned}$$

显然，$\beta_C^+(S) \leq \beta_C(S)$。

设 $C \in \{EAD, ESR\}$。记

$$\begin{aligned} \beta_C(S) &= \max\{\beta(I_C(S_\alpha)) : \alpha \in E_C(S)\}, \\ \beta_C^+(S) &= \max\{\beta^+(I_C(S_\alpha)) : \alpha \in E_C(S)\}. \end{aligned}$$

显然，$\beta_C^+(S) \leq \beta_C(S)$。

引理 14. 设 S 是一个满足 $n(S) > 0$ 且 $C \in \{AR, EAR, EAD, ESR\}$的决策规则系统。则 $h_C(S) \geq \beta_C(S) \geq \beta_C^+(S)$。

证明。设 $C \in \{AR, EAR\}$、$\alpha \in E_C(S)$ 和 $\beta_C(S) = \beta(S_\alpha)$。根据引理 4，我们得到$h_C(S) \geq h_C(S_\alpha)$。根据引理5，$h_C(S_\alpha) \geq \beta(S_\alpha) = \beta_C(S)$。正如我们之前所提到的，$\beta_C(S) \geq \beta_C^+(S)$。

设 $C \in \{EAD, ESR\}$、$\alpha \in E_C(S)$ 和 $\beta_C(S) = \beta(I_C(S_\alpha))$。根据引理 4，我们得到$h_C(S) \geq h_C(S_\alpha)$。根据引理5，$h_C(S_\alpha) \geq \beta(I_C(S_\alpha)) = \beta_C(S)$。正如我们之前所提到的，$\beta_C(S) \geq \beta_C^+(S)$。

□

引理 15. 设 S 是一个决策规则系统，具有 $n(S) > 0$ 和 $C \in \{AR, EAR\}$。那么

$$h_C(S) \leq d(S)\beta_C^+(S).$$

证明. 设 $V_C = V(S)$ 若 $C = AR$，且 $V_C = EV(S)$ 若 $C = EAR$。现在我们描述决策树 Γ 对来自 V_C的元组的处理过程。该过程由若干轮次组成。

第一轮。我们构造超图 $G(S^+)$ 的节点覆盖 B_1，使其基数最小，即 $|B_1| = \beta(S^+)$。决策树 Γ 依次计算来自 B_1的属性值。由此，我们得到一个由 $|B_1|$ 个方程组成的系统 α_1，这些方程的形式为 $a_{i_j} = \delta_j$，其中 $a_{i_j} \in B_1$，而 δ_j 是属性 a_{i_j}的计算值。如果 $S_{\alpha_1} = \emptyset$，或者 S_{α_1} 中的所有规则的左侧都为空，则树 Γ 结束其处理。处理结果是来自 S的决策规则集合 r，其中方程组 $K(r) \cup \alpha_1$相容。否则，我们继续进行决策树 Γ 处理的第二轮。

第二轮。我们构造超图 $G((S_{\alpha_1})^+)$ 的节点覆盖 B_2，使其基数最小，即 $|B_2| = \beta((S_{\alpha_1})^+)$。决策树 Γ依次计算 B_2 中属性的值。因此，我们得到一个由 $|B_2|$ 个方程组成的方程组 α_2。如果 $S_{\alpha_1 \cup \alpha_2} = \emptyset$ 或 $S_{\alpha_1 \cup \alpha_2}$ 中的所有规则都具有空左侧，那么树 Γ 就完成了工作。其结果是 S 中满足方程组 $K(r) \cup \alpha_1 \cup \alpha_2$ 相容的决策规则集合 r。否则，我们继续进行决策树 Γ 工作的第三轮，依此类推，直到得到空规则组，或得到其中所有规则都具有空左侧的规则组。

显然，$d(S) > d(S_{\alpha_1}) > d(S_{\alpha_1 \cup \alpha_2}) > \cdots$。因此，轮次不超过 $d(S)$。每轮中由 Γ 计算其取值的属性数量至多为 $\beta_C^+(S)$。因此，$h(\Gamma) \leq d(S)\beta_C^+(S)$。容易验证，$\Gamma$ 解决了问题$C(S)$。因此，$h_C(S) \leq d(S)\beta_C^+(S)$。

□

引理 16。 设 S 是一个具有 $n(S) > 0$ 和 $C \in \{EAD, ESR\}$的决策规则系统。那么

$$h_C(S) \leq d(I_C(S))\beta_C^+(S).$$

证明。 记作 $V_C = EV(S)$。现在我们来描述决策树 Γ 对来自V_C的一个元组的处理过程。如果 $I_C(S)$ 中的所有规则的左侧均为空，则决策树 Γ完成其处理。处理结果为决策规则集合 $I_C(S)$。否则，我们进入决策树处理的第一轮 Γ 。

我们构造超图 $G(I_C(S)^+)$ 的一个节点覆盖 B_1 ，使其具有最小基数，即 $|B_1| = \beta(I_C(S)^+)$。决策树 Γ 依次计算来自 B_1的属性值。由此，我们得到一个方程组 α_1 ，它由 $|B_1|$ 个形如 $a_{i_j} = \delta_j$ 的方程组成，其中 $a_{i_j} \in B_1$ ，而 δ_j 是属性 a_{i_j} 的计算值。如果 $I_C(S_{\alpha_1}) = \emptyset$，或者 $I_C(S_{\alpha_1})$ 中的所有规则的左侧均为空，则决策树 Γ完成其处理。处理结果为来自 S 的决策规则集合 r ，其中方程组 $K(r) \cup \alpha_1$ 相容，且 $A(r) \subseteq B_1$。否则，我们进入决策树处理的第二轮 Γ。

我们构造超图 $G(I_C(S_{\alpha_1})^+)$ 的一个具有最小基数的节点覆盖 B_2 ，即 $|B_2| = \beta(I_C(S_{\alpha_1})^+)$。决策树 Γ 按顺序计算来自 B_2 的属性值。因此，我们得到一个由 $|B_2|$ 个方程组成的方程组 α_2 。如果 $I_C(S_{\alpha_1 \cup \alpha_2}) = \emptyset$，或者来自 $I_C(S_{\alpha_1 \cup \alpha_2})$ 的所有规则都具有空左侧，那么决策树 Γ 就完成其工作。该工作的结果是来自 S 的决策规则集合 r ，对于这些规则，方程组 $K(r) \cup \alpha_1 \cup \alpha_2$ 是相容的，并且 $A(r) \subseteq B_1 \cup B_2$。否则，我们将继续进行决策树工作的第三轮 Γ ，依此类推，直到获得空规则系统，或者获得所有规则均具有空左侧的规则系统。

可以证明， $d(I_C(S)) > d(I_C(S_{\alpha_1})) > d(I_C(S_{\alpha_1 \cup \alpha_2})) > \cdots$。因此，轮次不超过 $d(I_C(S))$。在每一轮中由 Γ计算其值的属性数量至多为 $\beta_C^+(S)$。因此， $h(\Gamma) \leq d(I_C(S))\beta_C^+(S)$。可以证明，$\Gamma$ 解决了问题 $C(S)$。因此， $h_C(S) \leq d(I_C(S))\beta_C^+(S)$。 □

定理 4。 设 S 是一个决策规则系统，其中 $n(S) > 0$。则
(a) $\max\{d(S)\ \beta_{AR}(S)\} \leq h_{AR}(S) \leq d(S)\beta_{AR}^+(S)$。 (b) $\max\{d(S)\ \beta_{EAR}(S)\} \leq h_{EAR}(S) \leq d(S)\beta_{EAR}^+(S)$。 (c) $h_{AD}(S) \leq d(S)\beta_{AD}^+(S)$。 (d) $\max\{d(S')\ \beta_{EAD}(S')\} \leq h_{EAD}(S) \leq d(S')\beta_{EAD}^+(S')$，其中 $S' = R_{AD}(S)$。 (e) $h_{SR}(S) \leq d(S)\beta_{SR}^+(S)$。 (f) $\max\{d(S')\ \beta_{ESR}(S')\} \leq h_{ESR}(S) \leq d(S')\beta_{ESR}^+(S')$，其中 $S' = R_{SR}(S)$。

证明。 (a) $h_{AR}(S)$ 的下界由引理 6 和引理 14 得出。on$h_{AR}(S)$ 的上界由引理 15 得出。

(b) $h_{EAR}(S)$ 的下界由引理 6 和引理 14 得出。on$h_{EAR}(S)$ 的上界由引理 15 得出。

(c) 根据引理 15，可得 $h_{AR}(S) \leq d(S)\beta_{AR}^+(S)$。不等式 $h_{AD}(S) \leq h_{AR}(S)$ 可由引理 3 得出。显然， $\beta_{AD}^+(S) = \beta_{AR}^+(S)$。因此， $h_{AD}(S) \leq d(S)\beta_{AD}^+(S)$。

(d) 由引理 7 可知，$h_{EAD}(S) = h_{EAD}(S')$。这些界限

$$\max\{d(S'), \beta_{EAD}(S')\} \leq h_{EAD}(S')$$

可由引理 6 和引理 14 得出。界 $h_{EAD}(S') \leq d(I_{EAD}(S'))\beta^+_{EAD}(S')$ 可由引理 16 得出。显然，$I_{EAD}(S') = S'$。

(e) 根据引理 15 可得 $h_{AR}(S) \leq d(S)\beta^+_{AR}(S)$。不等式 $h_{SR}(S) \leq h_{AR}(S)$ 可由引理 3 得出。显然，$\beta^+_{SR}(S) = \beta^+_{AR}(S)$。因此，$h_{SR}(S) \leq d(S)\beta^+_{SR}(S)$。

(f) 根据引理 7，可得 $h_{ESR}(S) = h_{ESR}(S')$。这些界值

$$\max\{d(S'), \beta_{ESR}(S')\} \leq h_{ESR}(S')$$

由引理 6 和 14 得出。界限 $h_{ESR}(S') \leq d(I_{ESR}(S'))\beta^+_{ESR}(S')$ 由引理 16 得出。显然，$I_{ESR}(S') = S'$。 □

注释 3。注意，定理中提到的关于 $h_{EAR}(S)$ 和 $h_{ESR}(S)$ 的结果已发表于 [27]，但未附带证明。

7.2 算法

引理 15 和引理 16 的证明中描述的算法在实践中无法使用，因为它们需要构造一个基数最小的超图节点覆盖，而这是一个 NP难问题。本节中，我们将介绍一种用于构造节点覆盖的多项式时间算法，并对引理 15 和引理 16 的证明中描述的算法进行修改。

7.2.1 算法 $\mathcal{A}_{cover}$

设 S 是一个决策规则系统，满足 $n(S) > 0$，并设 S^+ 是其子系统，由 S 中长度为 $d(S)$ 的所有规则组成。现在我们描述一种多项式时间算法 $\mathcal{A}_{cover}$，用于构造超图 $G(S^+)$ 的节点覆盖 B，使得 $|B| \leq \beta(S^+)d(S)$。

算法 $\mathcal{A}_{cover}$

令 $B = \emptyset$。我们在 S^+ 中任意选择一条规则 r_1，并将 $A(r_1)$ 中的所有属性加入 B。从 S^+ 中移除所有满足 $A(r_1) \cap A(r) \neq \emptyset$ 的规则 r。将得到的系统记为 S^+_1。若 $S^+_1 = \emptyset$，则 B 是 $G(S^+)$ 的节点覆盖。若 $S^+_1 \neq \emptyset$，则我们在 S^+_1 中任意选择一条规则 r_2，并将 $A(r_2)$ 中的所有属性加入 B。从 S^+_1 中移除所有满足$A(r_2) \cap A(r) \neq \emptyset$ 的规则 r。将得到的系统记为 S^+_2。若 $S^+_2 = \emptyset$，则 B 是$G(S^+)$ 的节点覆盖。若 $S^+_2 \neq \emptyset$，则我们在 S^+_2 中任意选择一条规则 r_3，依此继续，直至构造出节点覆盖 B。

令 $B = \bigcup_{i=1}^{t} A(r_i)$。由于集合 $A(r_1) \ldots, A(r_t)$ 两两不相交，且每条规则 $r_1, \ldots, r_t$的长度都等于 $d(S)$，我们得到 $|B| = t \cdot d(S)$ 和 $\beta(S^+) \geq t$。因此$|B| \leq \beta(S^+)d(S)$。可以证明，该算法 $\mathcal{A}_{cover}$ 的时间复杂度为多项式。

7.2.2 算法 $\mathcal{A}^{C}_{tree}$, $C \in \{AR, EAR\}$

设 S 为一个决策规则系统，其中 $n(S) > 0, C \in \{AR, EAR\}, V_C = V(S)$ if $C = AR$, and $V_C = EV(S)$ if $C = EAR$。现在我们描述一个多项式时间算法 $\mathcal{A}^{C}_{tree}$：对于给定的、取自集合 V_C的属性值元组，该算法描述决策树 Γ对该元组的处理。该决策树解决问题 $C(S)$，并且满足 $h(\Gamma) \leq d(S)^2\beta^{+}_{C}(S)$。该算法是对引理 15 证明中所述算法的修改。

算法 $\mathcal{A}^{C}_{tree}$

决策树 Γ 的工作由多个轮次组成。

第一轮。使用算法 $\mathcal{A}_{cover}$，我们构造超图$G(S^{+})$的节点覆盖 B_1，其中包含 $|B_1| \leq \beta(S^{+})d(S)$。决策树 Γ 依次计算来自 B_1的属性值。由此，我们得到一个由 $|B_1|$ 个形如 $a_{i_j} = \delta_j$ 的方程组成的方程组 α_1，其中 $a_{i_j} \in B_1$，而 δ_j 是属性 a_{i_j} 的计算值。如果 $S_{\alpha_1} = \emptyset$ 或者来自 S_{α_1} 的所有规则都具有空左侧，那么树 Γ 就结束其工作。该工作的结果是来自 S 的决策规则集合 r，对于这些规则，方程组 $K(r) \cup \alpha_1$ 是相容的。否则，我们继续进行决策树 Γ 工作的第二轮。

第二轮。使用算法 $\mathcal{A}_{cover}$，我们构造超图$G((S_{\alpha_1})^{+})$的节点覆盖 B_2，其中包含 $|B_2| \leq \beta((S_{\alpha_1})^{+})d(S_{\alpha_1}) \leq \beta^{+}(S_{\alpha_1})d(S)$。决策树 Γ依次计算来自 B_2的属性值。由此，我们得到一个由$|B_2|$ 个方程组成的系统 α_2。如果 $S_{\alpha_1 \cup \alpha_2} = \emptyset$ 或来自 $S_{\alpha_1 \cup \alpha_2}$ 的所有规则都具有空左侧，那么树 Γ 便完成其工作。该工作的结果是来自 S 的决策规则集合 r，其中方程组 $K(r) \cup \alpha_1 \cup \alpha_2$ 相容。否则，我们继续进行决策树 Γ 工作的第三轮，依此类推，直到得到空规则系统，或得到一个所有规则都具有空左侧的系统。

显然，$d(S) > d(S_{\alpha_1}) > d(S_{\alpha_1 \cup \alpha_2}) > \cdots$。因此，轮次数至多为 $d(S)$。每轮中由 Γ 计算其值的属性数量至多为 $\beta^{+}_{C}(S)d(S)$。因此，$h(\Gamma) \leq d(S)^2\beta^{+}_{C}(S)$。容易验证，Γ 解决了问题 $C(S)$。可以证明，算法 $\mathcal{A}^{C}_{tree}$ 具有多项式时间复杂性。

设 S 是一个含有 $n(S) > 0$ 的决策规则系统。为简单起见，我们假设 $A(S) = \{a_1, \ldots, a_n\}$。现在，我们说明如何使用算法 $\mathcal{A}^{AR}_{tree}$ 和 $\mathcal{A}^{EAR}_{tree}$，来描述决策树在元组 $\delta = (\delta_1$、.....、$\delta_n)$ 的属性值上求解问题 $AR(S)$、$EAR(S)$、$AD(S)$ 和 $SR(S)$ 的过程。

问题 $AR(S)$。我们将算法 $\mathcal{A}^{AR}_{tree}$ 应用于决策规则系统 S 和元组 $\delta \in V(S)$。该算法描述了决策树 Γ 的运行过程；该树求解问题$AR(S)$，且 $h(\Gamma) \leq d(S)^2\beta^{+}_{AR}(S)$。根据引理 6，我们得到 $d(S) \leq h_{AR}(S)$。由引理 14 可知，$\beta^{+}_{AR}(S) \leq h_{AR}(S)$。因此，$h(\Gamma) \leq h_{AR}(S)^3$。

问题 $EAR(S)$。我们将算法 $\mathcal{A}^{EAR}_{tree}$应用于决策规则系统 S 和元组 $\delta \in EV(S)$。该算法描述了决策树 Γ的运行过程，该决策树解决了问题 $EAR(S)$，且满足 $h(\Gamma) \leq d(S)^2\beta^{+}_{EAR}(S)$。利用引理 6，可得 $d(S) \leq h_{EAR}(S)$。由引理 14可知，$\beta^{+}_{EAR}(S) \leq h_{EAR}(S)$。因此$h(\Gamma) \leq h_{EAR}(S)^3$。

问题 $AD(S)$。我们将算法 $\mathcal{A}^{AR}_{tree}$ 应用于决策规则系统 S 和元组 $\delta \in V(S)$。该算法描述了决策树 Γ的运行过程，它解决了问题

$AR(S)$，并且对于该问题， $h(\Gamma) \leq d(S)^2\beta^+_{AR}(S)$。利用引理 2，我们得到 Γ 解决了问题 $AD(S)$。显然， $\beta^+_{AR}(S) = \beta^+_{AD}(S)$。因此， $h(\Gamma) \leq d(S)^2\beta^+_{AD}(S)$。

问题 $SR(S)$。我们将算法 $\mathcal{A}^{AR}_{tree}$ 应用于决策规则系统 S 和元组 $\delta \in V(S)$。该算法描述了决策树 Γ的运行过程，它解决问题$AR(S)$，并且对于该问题， $h(\Gamma) \leq d(S)^2\beta^+_{AR}(S)$。利用引理 2，我们得到 Γ 解决问题 $SR(S)$。显然，$\beta^+_{AR}(S) = \beta^+_{SR}(S)$。因此， $h(\Gamma) \leq d(S)^2\beta^+_{SR}(S)$。

7.2.3 算法 $\mathcal{B}^C_{tree}$, $C \in \{ESR, EAD\}$

设 S 是一个决策规则系统，满足 $n(S) > 0$、 $C \in \{ESR, EAD\}$，且 $V_C = EV(S)$。现在我们描述一个多项式时间算法 $\mathcal{B}^C_{tree}$： 对于给定的、取自集合 V_C的属性值元组，该算法描述决策树 Γ在该元组上的运行过程。该决策树解决问题 $C(S)$，并且对于该问题， $h(\Gamma) \leq d(I_C(S))^2\beta^+_C(S)$。此算法是对引理 16 证明中所述算法的修改。

算法 $\mathcal{B}^C_{tree}$

如果 $I_C(S)$ 中的所有规则的左侧都为空，那么树 Γ 就完成了工作。这项工作的结果是决策规则集 $I_C(S)$。否则，我们将进入决策树 Γ 工作的第一轮。

使用算法 $\mathcal{A}_{cover}$，我们构造超图 $G(I_C(S)^+)$ 的节点覆盖 B_1，其中 $|B_1| \leq \beta(I_C(S)^+)d(I_C(S))$。决策树 Γ 依次计算来自 B_1 的属性值。因此，我们得到一个由 $|B_1|$ 个形如 $a_{i_j} = \delta_j$ 的方程组成的方程组 α_1，其中 $a_{i_j} \in B_1$ 和 δ_j 是属性 a_{i_j} 的计算值。如果 $I_C(S_{\alpha_1}) = \emptyset$，或者来自 $I_C(S_{\alpha_1})$ 的所有规则都具有空左侧，则树 Γ 结束其工作。该工作的结果是来自 S 的决策规则集合 r，对于这些规则，方程组 $K(r) \cup \alpha_1$ 是相容的，并且 $A(r) \subseteq B_1$。否则，我们继续进行决策树 Γ 工作的第二轮。

使用算法 $\mathcal{A}_{cover}$，我们构造超图 $G(I_C(S_{\alpha_1})^+)$ 的节点覆盖 B_2，其中包含 $|B_2| \leq \beta(I_C(S_{\alpha_1})^+)d(I_C(S_{\alpha_1})) \leq \beta(I_C(S_{\alpha_1})^+)d(I_C(S))$。决策树 Γ 依次计算来自 B_2 的属性值。由此，我们得到一个由 $|B_2|$ 个方程组成的方程组 α_2。如果 $I_C(S_{\alpha_1 \cup \alpha_2}) = \emptyset$，或者 $I_C(S_{\alpha_1 \cup \alpha_2})$ 中的所有规则都具有空左侧，那么树 Γ 就完成其工作。其工作的结果是来自 S 的决策规则集合 r，对于该集合，方程组 $K(r) \cup \alpha_1 \cup \alpha_2$ 相容，且 $A(r) \subseteq B_1 \cup B_2$。否则，我们继续进行决策树 Γ 工作的第三轮，依此类推，直到得到空规则系统，或得到一个所有规则都具有空左侧的规则系统。

可以证明， $d(I_C(S)) > d(I_C(S_{\alpha_1})) > d(I_C(S_{\alpha_1 \cup \alpha_2})) > \cdots$。因此，轮次不超过 $d(I_C(S))$。每轮中，Γ 计算出的属性值数量至多为 $\beta^+_C(S)d(I_C(S))$。因此， $h(\Gamma) \leq d(I_C(S))^2\beta^+_C(S)$。可以证明， Γ 解决了问题 $C(S)$。容易验证，算法 $\mathcal{B}^C_{tree}$ 的时间复杂度为多项式级。

设 S 是一个具有 $n(S) > 0$的决策规则系统。为简单起见，我们假设 $A(S) = \{a_1, \ldots, a_n\}$。现在我们说明如何使用算法 $\mathcal{B}^{EAD}_{tree}$和 $\mathcal{B}^{ESR}_{tree}$，描述决策树在属性值元组 $\delta = (\delta_1$、.....、 $\delta_n)$ 上解决问题 $EAD(S)$ 和 $ESR(S)$ 的工作过程。

问题 $EAD(S)$。构造规则系统 $S' = R_{AD}(S)$。我们将算法$\mathcal{B}_{tree}^{EAD}$应用于决策规则系统 S'，并对其进行调整，使其能够处理元组 $\bar{\delta} = (\delta_1, \ldots\ldots, \delta_n) \in EV(S)$。从算法 $\mathcal{B}_{tree}^{EAD}$的描述可知，它不会计算来自 $A(S) \setminus A(S')$ 的属性值。设算法 $\mathcal{B}_{tree}^{EAR}$需要计算属性 $a_i \in A(S')$ 的值。如果 $\delta_i \in EV_{S'}(a_i)$，则 $\mathcal{B}_{tree}^{EAD}$ 将正常运行。如果 $\delta_i \notin EV_{S'}(a_i)$，则 $\mathcal{B}_{tree}^{EAD}$ 将以与情况 $\delta_i = *$相同的方式运行。可以证明，调整后的算法 $\mathcal{B}_{tree}^{EAD}$描述了一棵决策树 Γ的工作过程，该决策树解决问题$EAD(S)$，且 $h(\Gamma) \leq d(I_{EAD}(S'))^2\beta_{EAD}^{+}(S')$。利用引理 6，我们得到$d(I_{EAD}(S')) \leq d(S') \leq h_{EAD}(S')$。由引理 14 可知，$\beta_{EAD}^{+}(S') \leq h_{EAD}(S')$。因此 $h(\Gamma) \leq h_{EAD}(S')^3$。由引理 7 可知，$h_{EAD}(S) = h_{EAD}(S')$。因此，$h(\Gamma) \leq h_{EAD}(S)^3$。

问题 $ESR(S)$。构造规则系统 $S' = R_{SR}(S)$。我们将算法$\mathcal{B}_{tree}^{ESR}$应用于决策规则系统 S'，并对其进行调整，使其能够处理元组 $\bar{\delta} = (\delta_1, \ldots, \delta_n) \in EV(S)$。从算法 $\mathcal{B}_{tree}^{ESR}$的描述可知，它不会计算 $A(S) \setminus A(S')$中的属性值。设算法 $\mathcal{B}_{tree}^{ESR}$需要计算属性 $a_i \in A(S')$的值。如果 $\delta_i \in EV_{S'}(a_i)$，那么 $\mathcal{B}_{tree}^{ESR}$将正常运行。如果$\delta_i \notin EV_{S'}(a_i)$，那么 $\mathcal{B}_{tree}^{ESR}$的运行方式将与情形 $\delta_i = *$相同。可以证明，调整后的算法 $\mathcal{B}_{tree}^{ESR}$描述了一棵决策树 Γ的运行过程，该决策树解决问题 $ESR(S)$，且满足 $h(\Gamma) \leq d(I_{ESR}(S'))^2\beta_{ESR}^{+}(S')$。利用引理 6，我们得到 $d(I_{ESR}(S')) \leq d(S') \leq h_{ESR}(S')$。由引理 14可知，$\beta_{ESR}^{+}(S') \leq h_{ESR}(S')$。因此，$h(\Gamma) \leq h_{ESR}(S')^3$。由引理 7可知，$h_{ESR}(S) = h_{ESR}(S')$。因此，$h(\Gamma) \leq h_{ESR}(S)^3$。

备注 4。 请注意，与上述提及的问题 $EAR(S)$ 和 $ESR(S)$ 类似的结果已在 [27] 中发表，但未提供证明。

8 结论

本文研究了如何为给定的规则系统构建决策树，以及如何构建表示决策树的无环决策图，并讨论了对于给定输入，不构建完整的决策树，而仅构建该树中计算路径的可行性。未来的研究将重点关注如何利用动态规划和贪心算法，为给定的决策规则系统构建决策树。

致谢

本出版物所报告的研究得到了阿卜杜拉国王科技大学（KAUST）的支持。

参考文献

[1] Abdelhalim, A., Traor´e, I., Nakkabi, Y.：使用 RBDT- 1 从规则创建决策树。计算智能 **32**(2), 216–239 (2016)

[2] Abdelhalim, A., Traor´e, I., Sayed, B.：RBDT-1：一种新的基于规则的决策树生成技术。载于：G. Governatori, J. Hall, A. Paschke（编），规则互操作与应用国际研讨会，RuleML 2009，美国内华达州拉斯维加斯，2009年11月5日至7日。会议论文集，计算机科学讲义，第5858卷，第108–121页。施普林格（2009）[3] AbouEisha, H., Amin, T., Chikalov, I., Hussain, S., Moshkov, M.：组合优化与数据挖掘的动态规划扩展，智能系统参考丛书，第146卷。施普林格（2019）[4] Alsolami, F., Azad, M., Chikalov, I., Moshkov, M.：多值决策决策表的决策树、抑制树及规则，智能系统参考丛书，第156卷。施普林格（2020年）[5] Blum, M., Impagliazzo, R.：通用预言机与预言机类（扩展摘要）。载于：第28届计算机科学基础年度研讨会，美国加利福尼亚州洛杉矶，1987年10月27日至29日，第118–126页。IEEE计算机学会（1987）[6] Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A.：数值数据的逻辑分析。《数学规划》**79**, 163–190 (1997)[7]Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., Mayoraz, E., Muchnik, I.B.：逻辑数据分析的实现。《IEEE知识与数据工程汇刊》**12**(2), 292–306 (2000)[8] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.：《分类与回归树》。Wadsworth and Brooks（1984）[9] Buhrman, H., de Wolf, R.：复杂性度量与决策树复杂性：综述。《理论计算机科学》**288**(1), 21–43 (2002)[10] Cao, H.E.C., Sarlin, R., Jung, A.：通过最大可满足性学习可解释的决策规则。《IEEE Access》**8**, 218180–218185 (2020)[11] Chikalov, I., Lozin, V.V., Lozina, I., Moshkov, M., Nguyen, H.S., Skowron, A., Zielosko, B.：《数据分析的三种方法——测试理论、粗糙集与逻辑数据分析》，智能系统参考丛书，第41卷。施普林格（2013）[12] Durdymyradov, K., Moshkov, M.：从决策规则系统导出的决策树深度界。arXiv:2302.07063 [cs.CC](2023)。URL https://doi.org/10.48550/ arXiv.2302.07063[13] F¨urnkranz, J., Gamberger, D., Lavrac, N.：《规则学习基础》。认知技术。施普林格（2012）[14]Gilmore, E., Estivill-Castro, V., Hexel, R.：更具可解释性的决策树。载于：H. Sanjurjo-Gonz´alez, I. Pastor-L´opez, P.G. Bringas, H. Quinti´an, E. Corchado（编）

混合人工智能系统 - 第16届国际会议，HAIS 2021，西班牙毕尔巴鄂，2021年9月22日至24日，会议论文集，计算机科学讲义，第12886卷，第280–292页。施普林格（2021）[15]Hartmanis, J., Hemachandra, L.A.：单向函数、鲁棒性与NP完全集合的非同构性。载于：第二届复杂性理论结构年会会议论文集，美国纽约州伊萨卡，康奈尔大学，1987年6月16日至19日。IEEE计算机学会（1987）[16] Imam, I.F., Michalski, R.S.：从决策规则学习决策树：比较研究的一种方法与初步结果。智能信息系统杂志**2**(3)，279–304（1993）[17] Imam, I.F., Michalski, R.S.：决策树应从示例还是决策规则中学习？载于：H.J. Komorowski, Z.W. Ras（编）《智能系统方法论》，第7届国际研讨会，ISMIS '93，挪威特隆赫姆，1993年6月15日至18日，会议论文集，计算机科学讲义，第689卷，第395–404页。施普林格（1993）[18] Imam, I.F., Michalski, R.S.：决策学习：FRD方法与一项比较研究。载于：Z.W. Ras, M. Michalewicz（编）《智能系统基础》，第9届国际研讨会，ISMIS '96，波兰扎科帕内，1996年6月9日至13日，会议论文集，计算机科学讲义，第1079卷，第428–437页。施普林格（1996）[19] Kaufman, K.A., Michalski, R.S., Pietrzykowski, J., Wojtusiak, J.：集成式多任务归纳数据库VINLEN：初始实现与早期结果。载于：S. Dzeroski, J. Struyf（编）《归纳数据库中的知识发现》，第5届国际研讨会，KDID 2006，德国柏林，2006年9月18日，经修订的精选论文与特邀论文，计算机科学讲义，第4747卷，第116–133页。施普林格（2006）[20] Michalski, R.S., Imam, I.F.：从决策规则学习面向问题的决策结构：AQDT-2系统。载于：Z.W. Ras, M. Zemankova（编）《智能系统方法论》，第8届国际研讨会，ISMIS '94，美国北卡罗来纳州夏洛特，1994年10月16日至19日，会议论文集，计算机科学讲义，第869卷，第416–426页。施普林格（1994）[21] Michalski, R.S., Imam, I.F.：关于决策结构学习。Fundam. Informaticae**31**(1)，49–64（1997）[22] Molnar, C.：《可解释机器学习：使黑箱模型变得可解释的指南》，第2版（2022）。网址：christophm.github.io/interpretable-ml-book/[23] Moshkov, M.：计算布尔函数的决策树深度。Fundam. Informaticae **22**(3)，203–215（1995）[24] Moshkov, M.：确定性与非确定性决策树复杂度的比较分析：全局方法。Fundam. Informaticae**25**(2)，201–214（1996）

[25] Moshkov, M.：决策树与决策规则系统之间的一些关系。载于：L. Polkowski, A. Skowron（编）《粗糙集与计算当前趋势》，第一届国际会议，RSCTC'98，波兰华沙，1998年6月22日至26日，会议论文集，计算机科学讲义，第1424卷，第499–505页。施普林格（1998）[26] Moshkov, M.：用于粗糙计算的确定性与非确定性决策树。Fundam. Informaticae 41(3)，301–311（2000）[27] Moshkov, M.：将决策规则系统转换为决策树（俄文）。载于：第七届离散数学及其应用国际研讨会会议论文集，俄罗斯莫斯科，2001年1月29日至2月2日，第1部分，第21–26页。莫斯科国立大学数学与力学系应用研究中心（2001）[28] Moshkov, M.：依据决策树和决策规则系统的复杂性对无限信息系统进行分类。Fundam. Informaticae54(4)，345–368（2003）[29] Moshkov, M.：确定性与非确定性决策树复杂度的比较分析：局部方法。载于：J.F. Peters, A. Skowron（编）《粗糙集论丛 IV》，计算机科学讲义，第3700卷，第125–143页。施普林格（2005）[30] Moshkov, M.：决策树时间复杂度。载于：J.F. Peters, A. Skowron（编）《粗糙集论丛 III》，计算机科学讲义，第3400卷，第244–459页。施普林格（2005）[31] Moshkov, M.：《确定性与非确定性决策树的比较分析》，智能系统参考丛书，第179卷。施普林格（2020）[32] Moshkov, M., Piliszczuk, M., Zielosko, B.：《粗糙集中的部分覆盖、约简与决策规则——理论与应用》，计算智能研究，第145卷。施普林格（2008）[33] Moshkov, M., Zielosko, B.：《组合机器学习——一种粗糙集方法》，计算智能研究，第360卷。施普林格（2011）[34] Pawlak, Z.：《粗糙集——数据推理的理论方面》，理论与决策文库：*D*辑，第9卷。Kluwer（1991）[35] Pawlak, Z., Skowron, A.：粗糙集基础。Inf. Sci.177(1)，3–27（2007）[36] Quinlan, J.R.：从决策树生成产生式规则。载于：J.P. McDermott（编）第十届国际人工智能联合会议会议论文集。意大利米兰，1987年8月23日至28日，第304–307页。Morgan Kaufmann（1987）[37] Quinlan, J.R.：《C4.5：机器学习程序》。Morgan Kaufmann（1993）[38] Quinlan, J.R.：简化决策树。Int. J. Hum. Comput. Stud. 51(2)，497–510（1999）

[39] Rokach, L., Maimon, O.：《决策树数据挖掘：理论与应用》，机器感知与人工智能丛书，第69卷。世界科学出版社（2007年）[40] Silva, A., Gombolay, M.C., Killian, T.W., Jimenez, I.D.J., Son, S.：应用于强化学习的可解释可微决策树优化方法。载于：S. Chiappa, R. Calandra（编），第23届人工智能与统计国际会议，AISTATS 2020，2020年8月26日至28日，线上 [意大利西西里岛巴勒莫],机器学习研究论文集，第108卷，第1855–1865页。PMLR（2020年）[41] Szydlo, T., Sniezynski, B., Michalski, R.S.：归纳数据库系统 VINLEN 中的规则到树的转换。载于：M.A. Klopotek, S.T. Wierzchon, K. Trojanowski（编），智能信息处理与网络挖掘，国际 IIS：IIPWM'05 会议论文集，2005年6月13日至16日在波兰格但斯克举行，软计算进展，第31卷，第496–500页。施普林格（2005年）[42] Tardos, G.：查询复杂度，或者为何难以通过随机预言机区分 $NP^A \cap coNP^A$ 与 P^A A? 组合学**9(4)**，385–392（1989年）