arXiv:2301.04086v2 [cs.FL] 8 Jun 2023

# 关于具有多个折扣因子的折扣和自动机的比较 ⋆

Udi Boker⋆⋆[0000−0003−4322−8892] 和 Guy Hefetz[0000−0002−4451−6581]

赖希曼大学，以色列赫兹利亚
udiboker@runi.ac.il, ghefetz@gmail.com

**摘要**。我们研究有限词与无限词上的非确定性折扣和自动机的比较问题。也就是说，研究如何检验自动机 $\mathcal{A}$ 和 $\mathcal{B}$ 是否满足：对于所有词 $w$，$\mathcal{A}(w) = \mathcal{B}(w)$ $\mathcal{A}(w) \leq \mathcal{B}(w)$，或者 $\mathcal{A}(w) < \mathcal{B}(w)$。

已知，当两个自动机具有相同的单一整数折扣因子时，这些问题是可判定的；而在所有其他情形下，可判定性仍未解决：单一折扣因子是非整数有理数时；每个自动机都可以具有多个折扣因子时；甚至当每个自动机都只有一个整数折扣因子、但两者的折扣因子不同时也是如此。

我们证明，比较具有多个折扣因子的折扣和自动机是不可判定的，即使这些折扣因子全都是整数；而若每个自动机各自具有一个（可能不同的）整数折扣因子，则比较它们是可判定的。为此，我们还提供了算法，用于针对给定的非确定性自动机 $\mathcal{N}$ 和确定性自动机 $\mathcal{D}$（二者各自具有一个可能不同的有理折扣因子），判断对于所有词 $w$，$\mathcal{N}(w) = \mathcal{D}(w)$、$\mathcal{N}(w) \geq \mathcal{D}(w)$ 或 $\mathcal{N}(w) > \mathcal{D}(w)$ 是否成立。

**关键词：** 折扣和自动机 · 比较 · 包含。

## 1 引言

布尔自动机的等价性和包含性检查，即检查 $L(\mathcal{A}) = L(\mathcal{B})$、$L(\mathcal{A}) \subseteq L(\mathcal{B})$ 或 $L(\mathcal{A}) \subset L(\mathcal{B})$ 是否成立，其中 $L(\mathcal{A})$ 和 $L(\mathcal{B})$ 分别是 $\mathcal{A}$ 和 $\mathcal{B}$ 所识别的语言，在自动机理论于各个领域的应用中发挥着核心作用，尤其是在形式化验证中（例如 [34,26,17,33,35,28]）。同样，定量自动机的比较将等价性和包含性检查扩展为：询问 $\mathcal{A}(w) = \mathcal{B}(w)$、$\mathcal{A}(w) \leq \mathcal{B}(w)$ 或 $\mathcal{A}(w) < \mathcal{B}(w)$ 是否对所有词 $w$ 成立。这对于将定量自动机理论应用于各个领域，尤其是定量形式化验证（例如 [15,14,21,11,27,3,5,22]），至关重要。

折扣求和是定量自动机理论（例如，[19,12,14,15]）中常用的一种赋值函数，也广泛应用于其他计算模型，例如博弈（例如，[37,4,1]）、马尔可夫决策过程（例如，[23,29,16]）以及

⋆ 这是与发表于 FoSSaCS 2023 会议论文集中的章节同名的完整版本 [7]。⋆⋆ 本研究得到以色列科学基金会 2410/22 号项目的资助。

强化学习（例如， [32,36]）中也采用折扣求和，因为它形式化了这样一种观念：眼前的奖励胜过遥远未来可能获得的奖励；同样，遥远未来可能出现的问题（例如反应式系统中的错误）也不如当前的问题令人担忧。

非确定性折扣和自动机（NDA）的转移带有有理数权重，并且具有固定的有理数折扣因子 $\lambda > 1$。有限或无限运行的值，是对转移权重进行折扣求和所得的结果：运行中的第 $i$ 个转移的权重除以 $\lambda^i$。有限或无限词的值，是自动机在该词上的运行的下确界值。因此，NDA 实现了一个从词到实数的函数。

NDA 并非总能确定化 [15], ，因为它们对基本代数运算不封闭 [8], ，而且其比较的可判定性尚未得到证明，这与若干长期悬而未决的问题有关 [9]。然而，将 NDA 限定为具有整数折扣因子 $\lambda \in \mathbb{N} \setminus \{0, 1\}$ ，便可得到一类稳健的自动机，它们对确定化和代数运算封闭，且比较问题可判定 [8]。

文献中研究了非确定性折扣自动机的各种变体，包括函数型、$k$ 值、概率型等 [21,20,13]。然而，直到最近，所有这些模型都仅限于使用单个折扣因子。这是对一般折扣求和范式的一项重要限制，因为该范式考虑多个折扣因子。例如，马尔可夫决策过程和折扣和博弈允许同一实体中存在多个折扣因子[23,4]。在 [6], 中，非确定性折扣自动机被扩展为非确定性多重折扣自动机，从而允许使用多个折扣因子，其中每次转移都可以使用不同的折扣因子。研究特别关注整数非确定性多折扣自动机，即仅使用整数折扣因子的此类自动机，并分析它们能否保留整数折扣非确定性自动机的良好性质。研究表明，这类自动机通常不具有确定化封闭性，也不具有代数运算封闭性；而其中一个受限子类——整洁型非确定性多折扣和自动机，其折扣因子的选择取决于迄今读入词的前缀——则确实保留了整数折扣非确定性自动机的良好性质。

虽然具有相同选择函数的整洁型非确定性多折扣和自动机之间的比较可在多项式空间内判定 [6], ，但一般整数非确定性多折扣自动机之间的比较是否可判定仍是一个开放问题$\mathcal{A}$ ，$\mathcal{B}$ 亦然。两个具有不同（单一）折扣因子的整数折扣非确定性自动机之间的比较是否可判定，甚至也尚无定论。

我们证明，对于给定的 NMDA $\mathcal{N}$和确定性 NMDA（DMDA） $\mathcal{D}$，即使二者都只有整数折扣因子，在有限词与无限词上，判定 $\mathcal{N} \equiv \mathcal{D}$ 以及 $\mathcal{N} \leq \mathcal{D}$是否成立都是不可判定的；而在有限词上，判定 $\mathcal{N} < \mathcal{D}$是否成立也不可判定。我们通过将双计数器机的停机问题归约到该问题，证明这一不可判定性结果。总体思路与 [18,2], 中的类似归约相同，但关键在于如何用整数 NMDA 模拟计数器。首先，折扣求和并不适合用于模拟计数器，因为在折扣情形下，当前的增量比相隔很远的减量影响大得多。然而，我们证明，多个折扣因子在某种意义上能够消除时间的影响，使得自动机具有这样的性质：无论一个字母出现在词中的何处，它对自动机值的影响都相同

。（参见引理 1 和图 3。）证明的另一项主要内容是说明如何以非确定性方式调整自动机权重和折扣因子，从而“检测”计数器当前的值是否为 0。（参见图 5、6、8 和 9。）

好的一面是，我们给出了算法，用于判定给定的非确定性折扣自动机 $\mathcal{N}$ 和确定性非确定性折扣自动机（确定性折扣自动机） $\mathcal{D}$（其有理折扣因子任意且可能各不相同）是否满足 $\mathcal{N} \equiv \mathcal{D}$、 $\mathcal{N} \geq \mathcal{D}$ 或 $\mathcal{N} > \mathcal{D}$ （定理 4）。我们的算法适用于有限词与无限词；当自动机的权重以二进制表示、折扣因子以一元表示时，算法的运行时间属于多项式空间。由于整数折扣非确定性自动机总是可以确定化 [8], ，我们的方法也提供了一种比较两个整数折扣非确定性自动机的算法，不过该算法未必属于多项式空间，因为确定化可能会使状态数量呈指数级增长。（尽管非确定性折扣自动机的确定化属于多项式空间 [8,6], ，但指数级的状态数可能使我们比较折扣因子不同的非确定性折扣自动机的算法需要指数级空间。）

比较折扣因子不同的自动机之所以困难，是因为它们不同的累积方式相结合后往往难以处理，导致比较整数非确定性多折扣自动机不可判定，也使比较有理折扣非确定性自动机以及分析非整数基下数的表示成为开放问题 [30,24,25,9]。然而，我们算法的核心观察是，当每个自动机都只有一个折扣因子时，我们可以将其计算树组合展开到某一层级 $k$，之后便可分别分析它们的延续：先处理折扣因子较低（衰减较慢）的自动机，再处理另一个自动机。其思路是，在层级 $k$之后，第二个自动机的折扣累积已显著得多，因此，即使第一个自动机发生一次非最优转移，也无法通过一个相对于第二个自动机更优的延续来弥补。因此，我们从层级 $k$开始计算第一个自动机的最优后缀词和运行，并在此基础上计算第二个自动机的最优运行。

## 2 预备知识

词。 字母表 $\Sigma$ 是一个任意的有限集合，而在 $\Sigma$ 上的词 是 $\Sigma$ 中字母构成的有限或无限序列，其中 $\varepsilon$ 表示空词。我们用 $u \cdot w$ 表示有限词 $u$ 与有限或无限词 $w$ 的连接，也可简写为 $uw$。我们将 $\Sigma^+$ 定义为除空词外所有有限词构成的集合，即 $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$。对于词 $w = \sigma_0\sigma_1\sigma_2\cdots$ 和索引 $i \leq j$，我们用 $w[i] = \sigma_i$ 表示索引 $i$ 处的字母，用 $w[i..j] = \sigma_i\sigma_{i+1}\cdots\sigma_j$ 表示从 $i$ 到 $j$ 的子词。

对于有限词 $w$ 和字母 $\sigma \in \Sigma$，我们用 $\#(\sigma, w)$ 表示 $w$ 中 $\sigma$ 出现的次数；对于集合 $S \subseteq \Sigma$，我们用 $\#(S, w)$ 表示 $\sum_{\sigma \in S} \#(\sigma, w)$。

对于有限字或无限字 $w$ 以及一个字母 $\sigma \in \Sigma$，我们将截至$w$ 为止的 $\sigma$前缀，$\mathrm{pref}_\sigma(w)$，定义为 $w$ 的最小前缀，其中包含一个字母，当

$w$ 中存在一个字母，或者如果 $w$ 本身不包含任何字母，则取其自身。形式上，$\mathsf{pref}_\sigma(w) = \begin{cases} w[0..\min\{i \mid w[i]=\sigma\}] & \text{若 } \sigma \text{ 在 } w \text{ 中出现} \\ w & \text{否则取最小值} w \end{cases}$

自动机。 非确定性折扣和自动机（NDA） [15] 是一种自动机，其转移具有有理数权重，并采用固定的有理数折扣因子 $\lambda > 1$。具有多个折扣因子的非确定性折扣和自动机（NMDA） [6] 与 NDA 类似，但每个转移可以采用不同的折扣因子。形式化定义如下：

**定义 1（[6]）**。有限词或无限词上的具有多个折扣因子的非确定性折扣和自动机（*NMDA*）是一个元组 $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$，其定义在字母表 $\Sigma$ 上，具有有限状态集 $Q$、初始状态集 $\iota \subseteq Q$、转移函数 $\delta \subseteq Q \times \Sigma \times Q$、权重函数 $\gamma\colon \delta \to \mathbb{Q}$，以及折扣因子函数 $\rho\colon \delta \to \mathbb{Q} \cap (1, \infty)$，该函数为每个转移指定其折扣因子，且该因子是大于 *1* 的有理数。 [1]

– $\mathcal{A}$ 的一次运行 是由状态和字母表字母组成的序列，$p_0, \sigma_0, p_1, \sigma_1, p_2, \cdots$，其中 $p_0 \in \iota$ 是一个初始状态，并且对于每个 $i$，$(p_i, \sigma_i, p_{i+1}) \in \delta$。– 运行 $r$的长度 记为 $|r|$；对于有限运行 $r = p_0, \sigma_0, p_1, ,$ 其长度为 $n$

$\cdots$、$\sigma_{n-1}$、$p_n$和 $\infty$ 用于无限运行。– 对于索引 $i < |r|$，我们将 $r$ 的第 $i$个转移定义为 $r[i] = (p_i, \sigma_i, p_{i+1})$，并将具有 $i$ 个转移的前缀运行定义为 $r[0..i] = p_0, \sigma_0, p_1, \cdots, \sigma_i, p_{i+1}$。– 有限/无限运行 $r$ 的值为 $\mathcal{A}(r) = \sum_{i=0}^{|r|-1} \left( \gamma(r[i]) \right) \cdot \prod_{j=0}^{i-1} \frac{1}{\rho(r[j])} \Big)$。例如，图 *1* 中 $\mathcal{A}$从 $\mathcal{A}$开始的运行 $r_1 = q_0, a, q_0, a, q_1, b, q_2$ 的值为 $\mathcal{A}(r_1) = 1 + \frac{1}{2} \cdot \frac{1}{3} + 2 \cdot \frac{1}{2 \cdot 3} = \frac{3}{2}$。– $\mathcal{A}$ 在有限或无限词 $w$ 上的值为$\mathcal{A}(w) = \inf\{\mathcal{A}(r) \mid r$ 是 $\mathcal{A}$ 在 $w\}$上的一次运行。– 对于每个有限运行 $r = p_0, \sigma_0, p_1, \cdots, \sigma_{n-1}, p_n$，我们将目标状态定义为 $\delta(r) = p_n$，将累积折扣因子定义为 $\rho(r) = \prod_{i=0}^{n-1} \rho(r[i])$。– 当所有折扣因子都是整数时，我们称 $\mathcal{A}$ 为整数 *NMDA*。– 若 $|\iota| = 1$，且对于每个 $q \in Q$ 和 $\sigma \in \Sigma$，都有$|\{q' \mid (q, \sigma, q') \in \delta\}| \leq 1$，则称 $\mathcal{A}$ 为确定性的，记作 *DMDA*，并将 $\delta$ 视为从词到状态的函数。– 当折扣因子函数 $\rho$ 为常数 $\rho \equiv \lambda \in \mathbb{Q} \cap (1, \infty)$时，我们称 $\mathcal{A}$ 为折扣因子为 $\lambda$ 的非确定性折扣和自动机（*NDA*） *[15]* （ *λ-NDA*）。如果 $\mathcal{A}$ 是确定性的，则它是一个 *λ-DDA*。– 对于状态 $q \in Q$，我们用 $\mathcal{A}^q$ 表示 *NMDA* $\mathcal{A}^q = \langle \Sigma, Q, \{q\}, \delta, \gamma, \rho \rangle$。

[1] 折扣因子有时定义为介于 0 和 1 之间的数；在这种定义下，权重乘以这些因子，而不是除以它们。

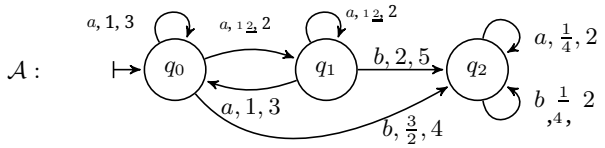


**图 1.** 一个 NMDA $\mathcal{A}$。转移上的标记表示字母表字母、转移的权重及其折扣因子。

计数器机。 双计数器机 [31] $\mathcal{M}$ 是由若干命令组成的序列$(l_1, \ldots, l_n)$，其中 $n \in \mathbb{N}$为某个值，涉及两个计数器 $x$ 和 $y$。我们将$\{1, \ldots, n\}$称为该机器的位置。对于每个 $i \in \{1, \ldots, n\}$，我们将$l_i$ 称为位置 $i$中的指令。命令有五种可能的形式:

$$\textsc{inc}(c),\ \textsc{dec}(c),\ \textsc{goto}\ l_k,\ \textsc{if}\ c{=}0\ \textsc{goto}\ l_k\ \textsc{else goto}\ l_{k'},\ \textsc{halt},$$

其中 $c \in \{x, y\}$是一个计数器， $1 \leq k$、 $k' \leq n$是位置。为避免将值为零的计数器 $c \in \{x, y\}$减小，每个 dec($c$) 指令前都先执行指令 if $c{=}0$ goto <当前_行> else goto <下一_行>，且没有其他直接跳转指令指向该位置。计数器最初均设为 0。图 2 给出了一个双计数器机的示例。

$l_1$. inc($x$)
$l_2$. inc($x$)
$l_3$. 如果 $x{=}0$ 则跳转到 $l_3$ ， 否则跳转到 $l_4$
$l_4$. dec($x$)
$l_5$。 如果 $x{=}0$ ， 转到 $l_6$ ， 否则转到 $l_3$
$l_6$。 停机

**图 2.** 双计数器机的一个示例。

设 $L$ 是 $\mathcal{M}$中可能的指令集合，则 $\mathcal{M}$ 的运行 是一个序列$\psi = \psi_1, \ldots, \psi_m \in (L \times \mathbb{N} \times \mathbb{N})^*$，满足以下条件:

1. $\psi_1 = \langle l_1, 0, 0\rangle$. 2. 对所有 $1 < i \leq m$， 令 $\psi_{i-1} = (l_j, \alpha_x, \alpha_y)$ 和 $\psi_i = (l', \alpha'_x, \alpha'_y)$。则有以下结论。– 若 $l_j$ 是 inc($x$) 指令（分别为 inc($y$)），则 $\alpha'_x = \alpha_x + 1, \alpha'_y = \alpha_y$（分别为 $\alpha_y = \alpha_y + 1, \alpha'_x = \alpha_x$），且 $l' = l_{j+1}$。– 若 $l_j$ 是 dec($x$)（分别为 dec($y$)），则 $\alpha'_x = \alpha_x - 1, \alpha'_y = \alpha_y$（分别为 $\alpha_y = \alpha_y - 1, \alpha'_x = \alpha_x$），且 $l' = l_{j+1}$。– 若 $l_j$ 是 goto $l_k$ ，则 $\alpha'_x = \alpha_x, \alpha'_y = \alpha_y$, 且 $l' = l_k$。– 若 $l_j$ 是 if $x{=}0$ goto $l_k$ else goto $l_{k'}$ ，则 $\alpha'_x = \alpha_x$, $\alpha'_y = \alpha_y$, 且当 $\alpha_x = 0$ 时 $l' = l_k$ ，否则 $l' = l_{k'}$。– 若 $l_j$ 是 if $y{=}0$ goto $l_k$ else goto $l_{k'}$ ，则 $\alpha'_x = \alpha_x, \alpha'_y = \alpha_y$, 且当 $\alpha_y = 0$ 时 $l' = l_k$ ，否则 $l' = l_{k'}$。– 若 $l'$ 是停机命令，则 $i = m$，也就是说，运行在停机后不会继续。

此外，若我们有 $\psi_m = \langle l_j, \alpha_x, \alpha_y \rangle$，且 $l_j$ 是停机命令，则称 $\psi$ 为停机运行。若机器 $\mathcal{M}$ 0的运行是停机运行，且结束于 $\langle l, 0, 0 \rangle$，则称该机器在 $\mathcal{M}$ 0处停机。若命令序列 $\tau \in L^*$ 符合运行 $\psi$，则表示 $\tau$ 是 $\psi$ 在其第一分量上的投影。

停机运行 $\psi = \psi_1, \ldots, \psi_m$ 的命令轨迹 $\pi = \sigma_1, \ldots, \sigma_m$ 描述了该运行的流程，包括描述每次遇到 if $c$=0 goto $l_k$ else goto $l_{k'}$ 指令时，计数器 $c$ 是等于 0 还是大于 0。其形式化定义如下。$\sigma_m =$ 停机，并且对于每个 $1 < i \leq m$，我们按照以下方式根据 $\psi_{i-1} = (l_j, \alpha_x, \alpha_y)$ 定义 $\sigma_{i-1}$：

- $\sigma_{i-1} = l_j$ 如果不是形如 if $c$=0 goto $l_k$ else goto $l_{k'}$ 的形式。- $\sigma_{i-1} =$ (goto $l_k, c = 0$) 对于 $c \in \{x, y\}$，若 $\alpha_c = 0$ 且指令 $l_j$ 形如 if $c$=0 goto $l_k$ else goto $l_{k'}$。- $\sigma_{i-1} =$ (goto $l_{k'}, c > 0$) 对于 $c \in \{x, y\}$，若 $\alpha_c > 0$ 且指令 $l_j$ 形如 if $c$=0 goto $l_k$ else goto $l_{k'}$。

例如，图 2 中机器的停机运行的命令轨迹为 inc($x$), inc($x$), (goto $l_4, x > 0$), dec($x$), (goto $l_3, x > 0$), (goto $l_4, x > 0$), dec($x$), (goto $l_6, x = 0$), halt。

判断给定的计数器机 $\mathcal{M}$ 是否停机，已知是不可判定的 [31]。判断 $\mathcal{M}$ 是否在两个计数器的值均为 0 时停机（称为 0-停机问题）同样不可判定。事实上，可以通过在每条停机命令之前添加一些清零计数器的命令，将停机问题归约为后者。

## 3 非确定性多重折扣自动机的比较

我们通过从双计数器机的停机问题进行归约，证明比较（整数）NMDA 是不可判定的。请注意，我们的 NMDA 仅使用整数折扣因子，但其权重并非整数。不过，只需将所有权重乘以一个公分母，并对计算作相应调整，就可以轻松地将权重也改为整数。

我们先给出一个关于某些折扣因子与权重级数累积值的引理。注意，根据该引理，无论折扣因子 $\lambda \in \mathbb{N} \setminus \{0, 1\}$ 与权重 $w = \frac{\lambda - 1}{\lambda}$ 在运行过程中出现在哪里，它们对累积值的影响都相同。这一性质将在用非确定性多重折扣自动机模拟计数时发挥关键作用。

**引理 1.** 对于每个由大于 1 的整数和权重组成的序列 $\lambda_1, \cdots, \lambda_m\, w_1, \cdots, w_m$，只要满足 $w_i = \frac{\lambda_i - 1}{\lambda_i}$，就有 $\sum_{i=1}^{m} \left( w_i \cdot \prod_{j=1}^{i-1} \frac{1}{\lambda_j} \right) = 1 - \frac{1}{\prod_{j=1}^{m} \lambda_j}$。

证明。 我们通过对 $m$ 进行归纳法来证明该断言。

基础情形，即 $m = 1$，是显然的。对于归纳步骤，我们有

$$\begin{aligned}\sum_{i=1}^{m+1}\left(w_i \cdot \prod_{j=1}^{i-1}\frac{1}{\lambda_j}\right) &= \sum_{i=1}^{m}\left(w_i \cdot \prod_{j=1}^{i-1}\frac{1}{\lambda_j}\right) + w_{m+1}\cdot\prod_{j=1}^{m}\frac{1}{\lambda_j}\\ &= 1 - \frac{1}{\prod_{j=1}^{m}\lambda_j} + \frac{\lambda_{m+1}-1}{\lambda_{m+1}}\cdot\prod_{j=1}^{m}\frac{1}{\lambda_j}\\ &= 1 - \frac{\lambda_{m+1}}{\prod_{j=1}^{m+1}\lambda_j} + \frac{\lambda_{m+1}-1}{\prod_{j=1}^{m+1}\lambda_j} = 1 - \frac{1}{\prod_{j=1}^{m+1}\lambda_j}\end{aligned}$$

□

### 3.1 归约

我们现在介绍从双计数器机的停机问题到 NMDA 包含问题的归约。我们给出针对有限词上的自动机的构造和正确性引理，然后在第 3.2 节说明如何将同一构造用于无限词上的自动机。

给定一台双计数器机 $\mathcal{M}$，其指令为 $(l_1, \ldots, l_n)$，我们构造一个整数 DMDA $\mathcal{A}$ 和一个在有限词上运行的整数 NMDA $\mathcal{B}$，使得 $\mathcal{M}$ 0 停机，当且仅当存在一个词 $w \in \Sigma^+$，满足 $\mathcal{B}(w) \geq \mathcal{A}(w)$；当且仅当存在一个词 $w \in \Sigma^+$，满足 $\mathcal{B}(w) > \mathcal{A}(w)$。

自动机 $\mathcal{A}$ 和 $\mathcal{B}$ 在以下字母表 $\Sigma$ 上运行，该字母表由 $5n+5$ 个字母组成，分别代表 $\mathcal{M}$ 的命令轨迹中可能出现的元素：

$$\begin{aligned}\Sigma^{\mathrm{INCDEC}} &= \{\,\mathrm{INC}(x), \mathrm{DEC}(x), \mathrm{INC}(y), \mathrm{DEC}(y)\,\}\\ \Sigma^{\mathrm{GOTO}} &= \{\mathrm{GOTO}\ \ l_k : k \in \{1,\ldots,n\}\}\cup\\ &\quad\ \{(\mathrm{GOTO}\ \ l_k, c=0) : k \in \{1,\ldots,n\}, c \in \{x,y\}\}\cup\\ &\quad\ \{(\mathrm{GOTO}\ \ l_{k'}, c>0) : k' \in \{1,\ldots,n\}, c \in \{x,y\}\}\\ \Sigma^{\mathrm{NOHALT}} &= \Sigma^{\mathrm{INCDEC}} \cup \Sigma^{\mathrm{GOTO}}\\ \Sigma &= \Sigma^{\mathrm{NOHALT}} \cup \{\mathrm{HALT}\}\end{aligned}$$

当 $\mathcal{A}$ 和 $\mathcal{B}$ 读取词 $w \in \Sigma^+$时，它们直观上会模拟一系列指令 $\tau_u$，该指令序列会产生命令轨迹 $u = \mathrm{prefhalt}(w)$。如果 $\tau_u$ 与 $\mathcal{M}$的实际运行相符，且该运行以 0 停机，那么 $\mathcal{B}$ 在 $w$ 上的最小运行的值严格大于 $\mathcal{A}(w)$。然而，如果 $\tau_u$ 与 $\mathcal{M}$的实际运行不相符，或者虽与实际运行相符但未以 0 停机，那么 $\mathcal{B}$ 会检测到这一违反情况；它在 $w$ 上的某次运行的值严格小于 $\mathcal{A}(w)$。

在构造中，我们使用以下部分折扣因子函数$\rho_p, \rho_d$：$\Sigma\text{nohalt} \to \mathbb{N}$，以及部分权重函数 $\gamma_p, \gamma_d$：$\Sigma\text{nohalt} \to \mathbb{Q}$。

$$\rho_p(\sigma) = \begin{cases} 5 & \sigma = \text{INC}(x) \\ 4 & \sigma = \text{DEC}(x) \\ 7 & \sigma = \text{INC}(y) \\ 6 & \sigma = \text{DEC}(y) \\ 15 & \text{otherwise} \end{cases} \qquad \rho_d(\sigma) = \begin{cases} 4 & \sigma = \text{INC}(x) \\ 5 & \sigma = \text{DEC}(x) \\ 6 & \sigma = \text{INC}(y) \\ 7 & \sigma = \text{DEC}(y) \\ 15 & \text{otherwise} \end{cases}$$

$\gamma_p(\sigma) = \frac{\rho_p(\sigma - 1)()}{\rho_p(\sigma)}$，以及 $\gamma_d(\sigma) = \frac{\rho_d(\sigma - 1)()}{\rho_d(\sigma)}$。我们称 $\rho_p$ 和 $\gamma_p$ 为原折扣因子函数和权重函数，而 $\rho_d$ 和 $\gamma_d$ 为对偶函数。注意，对于每个 $c \in \{x, y\}$，我们都有

$$\rho_p(\text{INC}(c)) = \rho_d(\text{DEC}(c)) > \rho_p(\text{DEC}(c)) = \rho_d(\text{INC}(c)) \tag{1}$$

直观地说，我们将使用原函数表示 $\mathcal{A}$ 的折扣因子和权重，并使用对偶函数识别违规情况。注意，如果沿着某次运行，将原函数改为对偶函数后，inc($c$) 字母出现的次数多于 dec($c$) 字母出现的次数，那么根据引理 1，该运行的值将低于原来的值。

下面继续给出它们的形式化定义。$\mathcal{A} = \langle \Sigma, \{q_\mathcal{A}, q^h_\mathcal{A}\}, \{q_\mathcal{A}\}, \delta_\mathcal{A}, \gamma_\mathcal{A}, \rho_\mathcal{A} \rangle$ 是一个整数 DMDA，由两个状态组成，如图 3 所示。注意，初始状态 $q_\mathcal{A}$ 对 $\Sigma$ 中的每个非停机字母都有自环，其权重和折扣因子根据原函数确定；此外还有一个转移 ($q_\mathcal{A}$, 停机$q^h_\mathcal{A}$), 其权重为 $\frac{14}{15}$，折扣因子为 15。

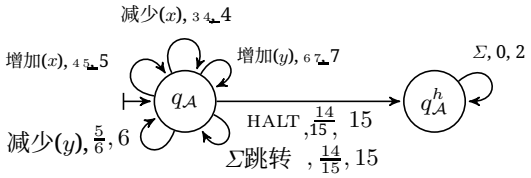


**图 3.** 为引理2的证明构造的 DMDA $\mathcal{A}$。

整数 NMDA $\mathcal{B} = \langle \Sigma, Q_\mathcal{B}, \iota_\mathcal{B}, \delta_\mathcal{B}, \gamma_\mathcal{B}, \rho_\mathcal{B} \rangle$ 是以下八个构件（检查器）的并集，每个构件负责检查对 $\mathcal{M}$ 的 0-停机运行描述中的某一类违规。它还包含状态 $q_{\text{freeze}}, q_{\text{halt}} \in Q_\mathcal{B}$，使得对于所有 $\sigma \in \Sigma$，都存在权重为 0 的转移（$q_{\text{freeze}}, \sigma, q_{\text{freeze}}$）$\in \delta_\mathcal{B}$和（$q_{\text{halt}}, \sigma, q_{\text{halt}}$）$\in \delta_\mathcal{B}$，其折扣因子任意。请注意，在 $\mathcal{B}$ 的所有构件中，字母 halt 上通向 $q_{\text{halt}}$ 的转移权重都高于 $\mathcal{A}$ 中相应转移的权重，因此，当未检测到违规时，$\mathcal{B}$ 在某个词上的值高于 $\mathcal{A}$ 在该词上的值。

**1. 停机检查器**。如图4所示，这个构件用于检查未停机运行中的违规情况。注意，其初始状态 $q_{\text{HC}}$ 的自环与 的初始状态的自环相同，

在遇到停机时转移至 $q_{\text{halt}}$，且权重高于 $\mathcal{A}$ 中对应的权重；在遇到每个非停机字母时，则转移至状态 $q_{\text{last}}$，以此“猜测”运行会在没有停机命令的情况下结束。

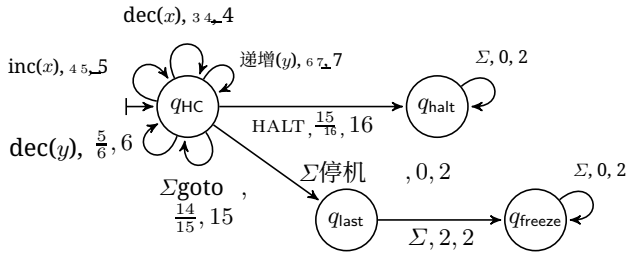


**图4.** NMDA $\mathcal{B}$中的停机检查器。

**2. 负计数器检查器。** 第二个构件如图5所示，用于检查输入前缀 $u$ 中对每个计数器 $c \in \{x, y\}$的 dec($c$) 命令数量是否不超过 inc($c$) 命令的数量。它与 $\mathcal{A}$ 类似，但其初始状态中包含自环；与 $\mathcal{A}$ 相比，这些自环更倾向于 dec($c$) 命令。

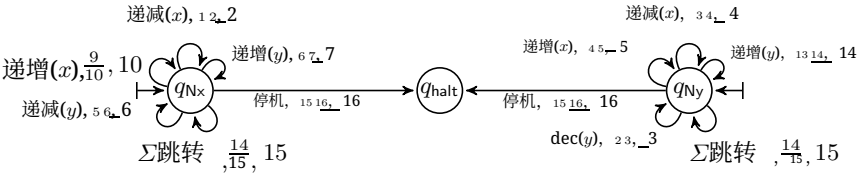


**图5.** NMDA $\mathcal{B}$中的负计数器检查器，左侧对应 $x$，右侧对应 $y$。

**3. 正计数器检查器。** 图6所示的第三个构件检查：对于每个 $c \in \{x, y\}$，输入前缀 $u$ 中的inc($c$)命令数都不多于dec($c$)命令数。它与 $\mathcal{A}$相似，但其初始状态中的自环对应的是对偶函数，而非原函数。

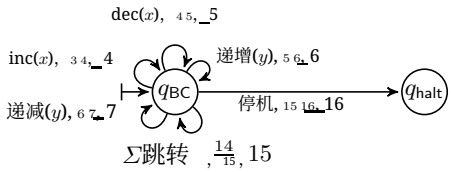


**图6.** NMDA $\mathcal{B}$ 中的正计数器检查器。

**4. 命令检查器。** 下一个构件用于检查连续命令中的局部违规。也就是说，它确保字母 $w_i$ 表示一条

可以跟在 $\mathcal{M}$中由 $w_{i-1}$ 表示的命令之后的命令，而不考虑计数器值。例如，如果位置 $l_2$ 处的命令是 inc($x$)，那么从与 $l_2$相关联的状态 $q_2$出发，我们沿着字母 inc($x$) 转移到与 $l_3$相关联的位置 $q_3$。该测试是局部的，因为这个构件不会检查因计数器值导致的非法跳转。图 2 中计数器机的命令检查器示例见图 7。

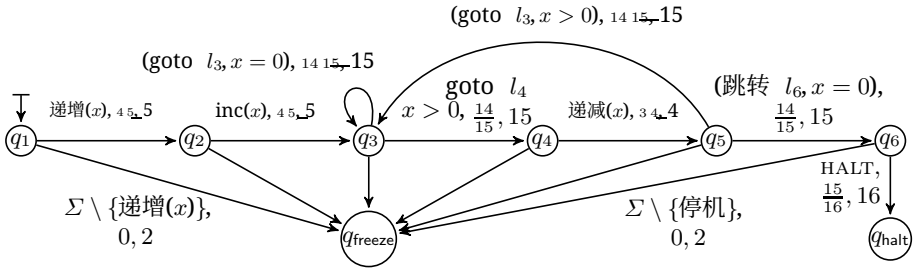


**图 7.** 与图 2 中计数器机对应的命令检查器。

命令检查器是一个 DMDA，由状态 $q_1$、.....、$q_n$ 组成，这些状态分别对应于命令 $l_1$、.....、$l_n$，以及状态 $q_{\text{halt}}$ 和 $q_{\text{freeze}}$。对于两个位置 $j$ 和 $k$，当且仅当在某个具有 $\sigma$（位于命令轨迹的相应位置）的 $\mathcal{M}$ 运行中，$l_k$ 可以在局部跟随 $l_j$ 时，才存在一条从 $q_j$ 到 $q_k$、标记为字母 $\sigma$ 的转移。也就是说，$l_j$ 是一个 goto $l_k$ 命令（即 $l_j = \sigma =$ goto $l_k$），$k$ 是 $j$ 之后的位置，而 $l_j$ 是一个 inc 或 dec 命令（即 $k = j + 1$ 和 $l_j = \sigma \in \Sigma$ incdec）；或者，$l_j$ 是一个 if $c$=0 goto $l_k$ else goto $l_{k'}$ 命令，其中 $\sigma =$（goto $l_k$，$c = 0$）；或者，$l_j$ 是一个 if $c$=0 goto $l_s$ else goto $l_k$ 命令，其中 $\sigma =$（goto $l_k$，$c > 0$）。上述 $\Sigma$ 个非停机转移的权重和折扣因子分别依据原函数 $\gamma_p$ 和 $\rho_p$ 确定。对于每个满足 $l_j =$ halt 的位置 $j$，都存在一条从 $q_j$ 到 $q_{\text{halt}}$ 的转移，该转移标记为字母 halt,_权重为 15 16，折扣因子为 16。其他所有未在上述内容中指定的转移都会导向 $q_{\text{freeze}}$，权重为 0，并带有某个折扣因子。

**5,6. 零跳转检查器**。接下来介绍的构件如图 8 所示，用于检查条件跳转中的违规情况。在这种情况下，我们为每个计数器 $c \in \{x, y\}$ 使用一个不同的检查器实例，确保对于每条 if $c$=0 goto $l_k$ else goto $l_{k'}$指令，如果跳转 goto $l_k$ 被执行，那么 $c$ 的值确实为 0。

直观地说，$q^c_{\mathsf{ZC}}$ 会从 inc($c$) 字母多于 dec($c$) 字母的词中获益，而 $q_c$ 则像 $\mathcal{A}$一样继续运行。如果在 inc($c$) 和 dec($c$) 的数量平衡后才转移到 $q_c$，就像真实命令轨迹中应有的那样，那么转移到 $q_c$之前的前缀词和之后的后缀词都不会带来收益。否则，只要运行结束时计数器为 0（负计数器和正计数器检查器可确保这一点），前缀词和后缀词都会带来收益，从而使运行的值更小。

**7,8. 正跳跃检查器**。这些构件如图9所示，与零跳转检查器互为对偶，用于检查条件跳转中的对偶违规情况。

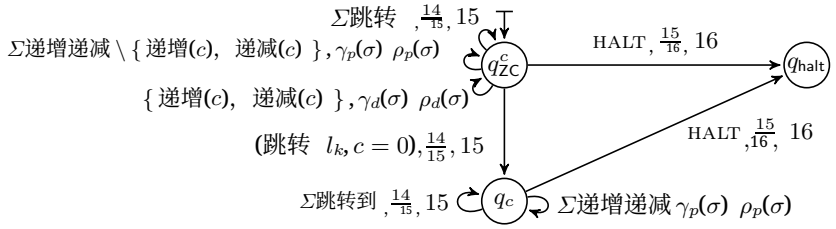


**图 8.** NMDA 中针对计数器 $c \in \{x, y\}$ 的零跳转检查器 $\mathcal{B}$。

与零跳转检查器类似，我们为每个计数器$c \in \{x, y\}$设置一个不同的实例，以确保对于每条if $c$=0 goto $l_k$ else goto $l_{k'}$ 指令，如果执行跳转goto $l_{k'}$，那么 $c$ 的值确实大于0。

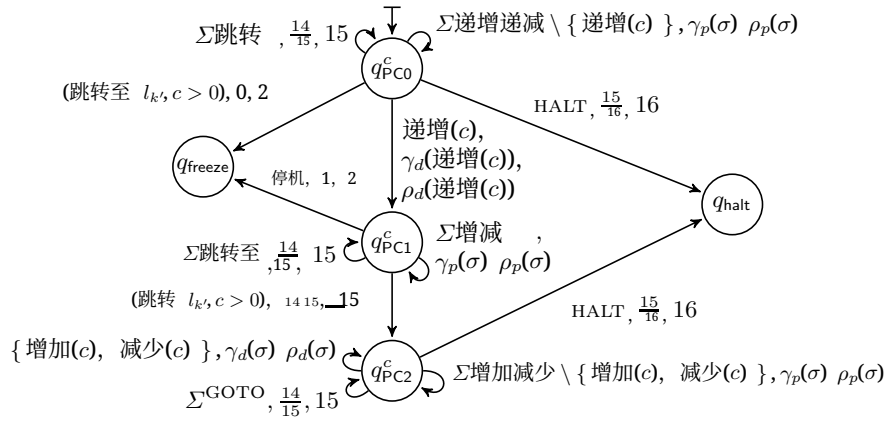


**图9**。 NMDA 中针对计数器 $c$ 的正跳跃检查器。

直观地说，如果在执行 (goto $l_{k'}, c > 0$) 指令时计数器为 0，且此前尚未执行 inc($c$) 指令，那么该构件可以通过从 $q^c_{\mathsf{PC0}}$ 移动到 $q_{\mathsf{freeze}}$ 获益。如果此前执行过 inc($c$) 指令，那么它可以通过在 inc($c$) 上从 $q^c_{\mathsf{PC0}}$ 移动到 $q^c_{\mathsf{PC1}}$ 时采用对偶函数，并在 dec($c$) 上对 $q^c_{\mathsf{PC1}}$ 再增加一个自循环来获益。

**引理2.** 给定一台双计数器机 $\mathcal{M}$，我们可以构造一个整数 *DMDA* $\mathcal{A}$ 和一个整数 *NMDA* $\mathcal{B}$，它们作用于有限词，使得 $\mathcal{M}$ 0 停机当且仅当存在一个词 $w \in \Sigma^+$，满足 $\mathcal{B}(w) \geq \mathcal{A}(w)$，当且仅当存在一个词 $w \in \Sigma^+$，使得$\mathcal{B}(w) > \mathcal{A}(w)$。

证明。给定一台双计数器机 $\mathcal{M}$，考虑第 3.1 节中构造的 DMDA $\mathcal{A}$ 和 NMDA$\mathcal{B}$，以及输入词 $w$。令 $u$ =prefhalt($w$)。

我们通过证明以下两点来证明该断言：I）如果 $u$ 正确描述了 $\mathcal{M}$ 的一次 0-停机运行，那么 $\mathcal{B}(w) > \mathcal{A}(w)$；II）如果 $u$ 与 $\mathcal{M}$的实际运行不符，或者

如果它与实际运行相符，但该运行没有 0-停机，那么 $\mathcal{B}$会检测到这一违规，具体来说， $\mathcal{B}(w) < \mathcal{A}(w)$。
**I.** 我们先考虑 $u$ 正确描述了 $\mathcal{M}$ 的一个 0-停机运行的情况，并证明 $\mathcal{B}(w) > \mathcal{A}(w)$。

注意，在 $\mathcal{B}$ 的所有检查器中，经过停机命令转移到 $q_{\mathsf{halt}}$ 状态的权重都高于 $\mathcal{A}$ 中对应转移的权重。因此，如果某个检查器在 $u$ 上的行为与 $\mathcal{A}$ 相同，即使用原函数，那么它产生的值会高于 $\mathcal{A}$ 的值。

下面我们证明，即使检查器非确定性地“猜测存在违规”，其行为不同于 $\mathcal{A}$，每个检查器产生的值仍高于 $\mathcal{A}$ 在 $u$ 上的值（这也等于 $\mathcal{A}$ 在 $w$ 上的值）。

*1.* 停机检查器。由于 $u$ 确实包含停机命令，因此，若停机检查器在 $u$ 上的运行猜测存在违规，就会以折扣因子 2，沿着从 $q_{\mathsf{HC}}$ 到 $q_{\mathsf{last}}$ 再到 $q_{\mathsf{freeze}}$ 的一对转移结束，其权重分别为 0 和 2。

令 $D$ 为构件中截至这对转移的累积折扣因子。根据引理 1，此时累积权重为$1 - \frac{1}{D}$，因此运行的值将为 $1 - \frac{1}{D} + \frac{1}{D} \cdot 0 + \frac{1}{2D} \cdot 2 = 1$；根据引理 1，该值大于在任意词上 $\mathcal{A}$ 的值。

*2,3.* 负计数器和正计数器检查器。由于 $u$ 中 inc($c$) 和 dec($c$) 字母的数量相同，根据公式（1）和 $Lemma\ 1$，这些构件与 $\mathcal{A}$ 在 $u$ 的前缀（截至最后一次转移）上的值相同；在最后一次转移上，这些构件的权重会更高。

*4.* 命令检查器。由于该构件是确定性的，因此它无法“猜测某项违规”，并且由于停机命令上的权重，其在 $u$ 上的值大于 $\mathcal{A}(u)$。

*5,6.* 零跳转检查器。考虑计数器 $c \in \{x, y\}$ 以及构件在 $u$上的一次运行 $r$。如果 $r$ 没有移至 $q_c$，那么与负计数器和正计数器检查器中的分析类似，我们有 $\mathcal{B}(r) > \mathcal{A}(w)$。否则，将 $r$ 移至 $q_c$ 时使用的转移记为 $t$。注意，由于 $u$ 与 $\mathcal{M}$的实际运行相对应，因此在 $c = 0$时确实执行了 $t$。在这种情况下，运行的值不会受到影响，因为在 $t$ 之前，inc($c$) 和 dec($c$) 字母的数量相同，而在 $t$ 之后，inc($c$) 和 dec($c$) 字母的数量也相同。因此，根据停机命令上的最后一次转移，我们有 $\mathcal{B}(r) > \mathcal{A}(u)$。

*7,8.* 正跳跃检查器。考虑计数器 $c \in \{x, y\}$ 以及构件在 $u$上的一次运行 $r$。如果 $r$ 始终无法到达 $q^c_{\mathsf{PC1}}$，那么除值更高的停机转移外，它与 $\mathcal{A}$具有相同的权重序列和折扣因子。如果 $r$ 到达$q^c_{\mathsf{PC1}}$，但始终无法到达 $q^c_{\mathsf{PC2}}$，由于 $u$ 以停机字母结尾，因此 $r$ 以一条指向 $q_{\mathsf{freeze}}$、权重为 1 的转移结尾，从而有 $\mathcal{B}(r) = 1 > \mathcal{A}(w)$。

如果 $r$ 到达 $q^c_{\mathsf{PC2}}$，则令 $u = y \cdot \mathrm{inc}(c) \cdot z \cdot v$，其中 $y$ 不含 inc($c$) 字母；$t = r[|y|+1+|z|]$ 是 $r$ 中以 $q^c_{\mathsf{PC2}}$为目标的第一条转移，而 $\alpha_c \geq 1$ 是执行 $t$ 时计数器 $c$ 的值。我们有 $1+\#(\mathrm{inc}(c)\ z) = \#(\mathrm{dec}(c)\ z)+\alpha_c$。由于 $u$是平衡的，我们还有 $\#(\mathrm{dec}(c)\ v) = \#(\mathrm{inc}(c)\ v)+\alpha_c$。对于

第一个 inc($c$) 字母， $r$ 获得的折扣因子为 $\rho_d$(inc($c$)) $= \rho_p$(dec($c$))。此后所有的 inc($c$) 和 dec($c$) 字母都依据 $z$ 中的 $\rho_p$以及 $v$中的 $\rho_d$ 来贡献折扣因子。因此， $r$ 获得折扣因子 $\rho_p$(dec($c$)) 的次数总计为

$$\begin{aligned} 1 + \#(\textsc{dec}(c), z) + \#(\textsc{inc}(c), v) &= 1 + 1 + \#(\textsc{inc}(c), z) - \alpha_c + \#(\textsc{inc}(c), v) \\ &= \#(\textsc{inc}(c), u) + 1 - \alpha_c \\ &\leq \#(\textsc{inc}(c), u) = \#(\textsc{dec}(c), u) \end{aligned}$$

次，折扣因子 $\rho_p$(inc($c$)) 总计

$$\begin{aligned} \#(\textsc{inc}(c), z) + \#(\textsc{dec}(c), v) &= \#(\textsc{inc}(c), z) + \#(\textsc{inc}(c), v) + \alpha_c \\ &= \#(\textsc{inc}(c), u) - 1 + \alpha_c \geq \#(\textsc{inc}(c), u) \end{aligned}$$

次。

因此，在停机转移之前， $r$ 在 $u$ 前缀上的值至少与 $\mathcal{A}$ 的值一样大；由于后者的 $r$ 权重更高，我们有 $\mathcal{B}(r) > \mathcal{A}(u)$。

**II.** 接下来考虑 $u$ 未能正确描述 $\mathcal{M}$的 0-停机运行的情况，并证明 $\mathcal{B}(w) < \mathcal{A}(w)$。注意，错误必然属于以下情况之一；与 $u$上的 $\mathcal{A}$ 的值相比，每种情况都会导致 $u$上 $\mathcal{B}$的某个构件的值较低：

– 词 $u$ 没有 停机指令。 在这种情况下，停机检查器在 $u$ 上的取最小值的运行，与其在 $\mathcal{A}$ 上的运行相同，直到最后一次转移；在该转移上，停机检查器的权重为 0，而在 $\mathcal{A}$上则为严格正值。

– 词 $u$ 描述的运行不会使两个计数器的值最终都变为 $0$ 。因此有以下几种子情况：

- 词 $u$ 中 对于某个计数器 $c \in \{x, y\}$，dec($c$) 字母的数量多于 inc($c$)字母的数量。对于 $c = x$，在负计数器检查器中，与 $\mathcal{A}$ 中的值相比，折扣因子从 4 变为 2 的数量多于从 5 变为 10 的数量，这意味着构件从开始到最后一个字母的总值将低于 $\mathcal{A}$ 在其上的值。对于 $c = y$，针对折扣因子 6;3 和 7;14，也有类似的分析。
- 词 $u$ 中 对于某个计数器 $c \in \{x, y\}$，inc($c$) 字母的数量多于 dec($c$)字母的数量。根据公式（1）和 $Lemma\ 1$，正计数器检查器从开始到最后一次转移的值将低于 $\mathcal{A}$ 从开始到最后一次转移的值。

不过请注意，停机转移 (16) 上构件的权重仍高于 $\mathcal{A}$ 在该转移上的权重 (15)。然而，由于一次“违规检测”会导致至少一个折扣因子从 4 替换为 2、从 6 替换为 3、从 5 替换为 4，或从 7 替换为 6（并替换相应的权重，以保持 $\frac{\rho-1}{\rho}$ 比率），而 16 与 15 之间的比率差异小于其他权重对之间的差异，因此构件的值，以及 $\mathcal{B}$ 在 $u$ 上的值，小于 $\mathcal{A}(u)$。事实上，根据

引理 1 $\mathcal{A}(u) = 1 - \frac{1}{D_{\mathcal{A}}}$，其中 $D_{\mathcal{A}}$ 是沿 $\mathcal{A}$ 的运行的折扣因子的乘积，而 $\mathcal{B}(u) \leq 1 - (\frac{1}{D_{\mathcal{A}}} \cdot \frac{7}{6} \cdot \frac{15}{16}) < 1 - \frac{1}{D_{\mathcal{A}}} = \mathcal{A}(u)$。

– 词 $u$ 无法正确描述 $\mathcal{M}$ 的运行。因此有以下子情况：

- 错误描述与条件跳转无关。此时，命令检查器在错误描述之前的前缀 $u$ 上，具有与 $\mathcal{A}$ 相同的权重和折扣因子；之后其权重均为 0，而 $\mathcal{A}$ 的权重严格为正。
- 错误描述与条件跳转有关。此时有以下子子情况：

  * 在 $\mathcal{M}$ 的运行中，位置 $i$ 处的计数器 $c > 0$，而 $u[i] =$ 跳转到 $l_k$，$c =$0。令 $v = u[0..i-1]$ 和 $u = v \cdot v'$，并考虑零跳转检查器在 $u$ 上的运行 $r$，它在 $v$ 之后转移到 $q_c$。那么 #(inc($c$) $v$) >#(dec($c$) $v$)，且 #(inc($c$) $v'$) <#(dec($c$) $v'$)。（我们可以假设 inc($c$) 和 dec($c$) 字母的总数相同，否则之前的某个检查器会检测到这一点。） $r[0..i-1]$ 中所有 inc($c$) 和 dec($c$) 转移的权重和折扣因子均由对偶函数决定，而 $r[i..|w|-1]$ 中的这些转移的权重和折扣因子则由原函数决定。因此，与 $\mathcal{A}$ 相比，从 $\gamma_p$(inc($c$)) 变为 $\gamma_d$(inc($c$)) $= \gamma_p$(dec($c$)) 的权重多于从 $\gamma_p$(dec($c$)) 变为 $\gamma_d$(dec($c$)) $= \gamma_p$(inc($c$)) 的权重，导致 $r$ 在 $u$ 上的总值低于 $\mathcal{A}$。（如负计数器和正计数器检查器的论证所示，停机转移较高的权重不如上述较低的值重要。）

  * 一个计数器 $c = 0$ 在.....的位置 $i$ 上 $\mathcal{M}$的运行中，而 $u[i] =$ 转到 $l_k, c >$ 0。设 $r$ 是正跳跃检查器在 $u$上的取最小值的运行。如果在位置 $i$之前，$u$ 中没有 inc($c$) 字母，那么直到 $i$的字母为止，$r$ 的权重和折扣因子都与 $\mathcal{A}$ 相同；在该字母处，它将从 $q^c_{\mathsf{PC1}}$ 转到 $q_{\text{冻结}}$，之后继续进行权重为 0 的转移；相比之下，$\mathcal{A}$中的对应转移权重严格为正。
    否则，$u$ 中的第一个 inc($c$) 字母会使 $r$ 从$q^c_{\mathsf{PC0}}$ 转到 $q^c_{\mathsf{PC1}}$，其折扣因子为 $\rho_d$(inc($c$))。那么在 $q^c_{\mathsf{PC1}}$ 中，dec($c$) 转移的数量多于 inc($c$) 转移的数量；而在 $q^c_{\mathsf{PC2}}$ 中，dec($c$) 和 inc($c$) 转移的数量相同。（我们可以假设 $u$ 通过了先前的检查器，因此 inc($c$) 字母和 dec($c$) 字母的总数相同。）因此，$\rho_d$(inc($c$)) 的折扣因子比 $\rho_p$(inc($c$)) 多两个，所得值小于 $\mathcal{A}(u)$。（与前面的情况一样，停机转移的较高值影响较小。）

□

### 3.2 比较的不可判定性

对于有限词，这一不可判定性结果可直接由引理2和计数器机的 0-停机问题的不可判定性推出 [31]。

**定理 1**。 在有限词上，整数非确定性多重折扣自动机之间的严格包含和非严格包含问题均不可判定。更确切地说，对于给定的整数非确定性多重折扣自动机 $\mathcal{N}$ 和整数 *DMDA* $\mathcal{D}$，判定是否对所有有限词 $w$都有 $\mathcal{N}(w) \leq \mathcal{D}(w)$，以及是否对所有有限词 $w$都有 $\mathcal{N}(w) < \mathcal{D}(w)$，这两个问题均不可判定。

对于无限词，非严格包含的不可判定性也可由第 3.1 节给出的归约推出，因为该归约会考察词中直到第一个停机命令为止的前缀。严格包含对于无限词是否也不可判定，这个问题仍未解决。后一个问题的难点在于：一个错误描述双计数器机停机运行的无限词 $w$ 中，可能永远不会出现停机命令；在这种情况下，归约中的两个自动机$\mathcal{A}$ 和 $\mathcal{B}$ 在 $w$上的值相同。对于含有停机命令、但未能正确描述双计数器机停机运行的词 $w$，有 $\mathcal{B}(w) < \mathcal{A}(w)$；而对于确实正确描述停机运行的词 $w$，有 $\mathcal{B}(w) > \mathcal{A}(w)$。因此，该归约仅能说明对于所有词 $w$，$\mathcal{B}(w) \leq \mathcal{A}(w)$是否成立，却不能说明对于所有词 $w$，$\mathcal{B}(w) < \mathcal{A}(w)$是否成立。

**定理 2.** 在无限词上，（整数）非确定性多重折扣自动机的非严格包含问题不可判定。更准确地说，对于给定的整数 *NMDA* $\mathcal{N}$和整数 *DMDA* $\mathcal{D}$，判定对所有无限词 $w$是否都有 $\mathcal{N}(w) \leq \mathcal{D}(w)$ 的问题不可判定。

证明. 第 3.1 节中归约所用的自动机 $\mathcal{A}$ 和 $\mathcal{B}$ 可原样用于无限词，只需忽略 $\mathcal{B}$ 的停机检查器组件，因为它仅与有限词相关。

由于 $\mathcal{A}$ 和 $\mathcal{B}$ 在输入词 $w$ 上的值只与 $w$ 的前缀 $u = \text{prefhalt}(w)$ 有关，该前缀截至第一个停机命令，因此，如果 $u$ 正确描述了双计数器机 $\mathcal{M}$ 的一次停机运行，我们仍有 $\mathcal{B}(w) > \mathcal{A}(w)$；如果 $u$ 是有限词，且未正确描述 $\mathcal{M}$的一次停机运行，则有 $\mathcal{B}(w) < \mathcal{A}(w)$。

然而，对于无限词，还可能出现词 $w$ 不包含停机命令的情况。此时，$\mathcal{A}$ 和 $\mathcal{B}$ 的命令检查器的值都会收敛到 1，从而得到 $\mathcal{A}(w) = \mathcal{B}(w)$。

因此，如果 $\mathcal{M}$ 0 停机，则存在一个词 $w$，使得 $\mathcal{B}(w) > \mathcal{A}(w)$；否则，对于所有词 $w$，都有 $\mathcal{B}(w) \leq \mathcal{A}(w)$。□

注意，对于 NMDA，等价性与非严格包含不可相互归约。

**定理 3.** 在有限词和无限词上，*(*整数*)* *NMDA* 的等价性都是不可判定的。也就是说，对于给定的整数 *NMDA* $\mathcal{A}$和$\mathcal{B}$，要判定在有限词或无限词上，对所有词 $w$ 是否有 $\mathcal{A}(w) = \mathcal{B}(w)$，这一问题不可判定。

证明。反设存在一个用于检查 $\mathcal{A}$ 和 $\mathcal{B}$等价性的过程。我们可以利用非确定性构造一个自动机$\mathcal{C} = \mathcal{A} \cup \mathcal{B}$，使得对于所有词 $w$，均有 $C(w) \leq A(w)$。随后，我们可以检查 $\mathcal{C}$是否等价于 $\mathcal{A}$；当且仅当对于所有词 $w$，均有 $\mathcal{A}(w) \leq \mathcal{B}(w)$ 时，这一等价关系成立。事实上，若 $\mathcal{A}(w) \leq \mathcal{B}(w)$，则 $\mathcal{A}(w) \leq \min(\mathcal{A}(w)\ \mathcal{B}(w)) = \mathcal{C}(w)$；而若存在一个词 $w$，使得 $\mathcal{B}(w) < \mathcal{A}(w)$，则有 $\mathcal{C}(w) = \min(\mathcal{A}(w)\ \mathcal{B}(w)) < \mathcal{A}(w)$，这意味着 $\mathcal{C}$ 和 $\mathcal{A}$ 并不等价。因此，这样的过程与定理1和定理2中所证明的非严格包含问题不可判定性相矛盾。□

## 4 不同折扣因子的非确定性折扣自动机比较

下面给出我们用于比较折扣因子不同的非确定性折扣自动机的算法。我们先讨论无限词上的自动机，然后说明如何通过归约到无限词的情形来解决有限词的情形。

该算法基于我们的主要观察：由于折扣因子不同，我们只需考虑自动机计算树组合到某个层级 $k$；此后，可以先考虑折扣因子较小的自动机的最佳/最差延续，再在此基础上考虑第二个自动机的最差/最佳延续。

对于一个非确定性折扣自动机 $\mathcal{A}$，我们将其最低（分别为最高）无限运行值定义为 lowrun($\mathcal{A}$)（分别为 highrun($\mathcal{A}$)）$=$ min（分别为 max）$\{\mathcal{A}(r) \mid r$ 是 $\mathcal{A}$ （在某个词 $w \in \Sigma^\omega$ 上）的无限运行$\}$。

注意，我们可以使用 min 和 max（而非 inf 和 sup），因为下确界和上确界确实能由该非确定性折扣自动机的特定无限运行取得（参见 [10, 定理证明 9]）。请注意，lowrun($\mathcal{A}$) 和 highrun($\mathcal{A}$) 可以通过简单归约到单人折扣收益博弈，在多项式时间内计算 [4]。

考虑词的值时，我们还将 $\mathcal{A}$的最低（分别是最高）字值称为 lowword($\mathcal{A}$)（分别是 highword($\mathcal{A}$)）$=$ min（分别是 max）$\{\mathcal{A}(w) \mid w \in \Sigma^\omega\}$。注意，lowword($\mathcal{A}$) $=$lowrun($\mathcal{A}$), highword($\mathcal{A}$) $\leq$highrun($\mathcal{A}$)，且对于确定性自动机，highword($\mathcal{A}$) = highrun($\mathcal{A}$)。

对于一个具有状态 $Q$的 NMDA $\mathcal{A}$，我们将 $\mathcal{A}$的后缀运行之间的最大差值定义为 maxdiff($\mathcal{A}$) $=$max $\{$highrun($\mathcal{A}^q$) $-$ lowrun($\mathcal{A}^q$) $\mid q \in Q\}$。注意，maxdiff($\mathcal{A}$) $\geq 0$，且 $\mathcal{A}^q(w)$有如下界限。

$$\text{LOWRUN}(\mathcal{A}^q) \leq \mathcal{A}^q(w) \leq \text{LOWRUN}(\mathcal{A}^q) + \text{MAXDIFF}(\mathcal{A}) \tag{2}$$

**引理 3.** 存在一个算法，对于每组输入折扣因子$\lambda_A, \lambda_D \in \mathbb{Q} \cap (1, \infty)$，都能计算 $\lambda_A$-非确定性折扣自动机 $\mathcal{A}$ 和 $\lambda_D$-确定性折扣自动机 $\mathcal{D}$ 在无限词上的值$\min\{\mathcal{A}(w) - \mathcal{D}(w) \mid w \in \Sigma^\omega\}$。

证明。考虑字母表 $\Sigma$、折扣因子 $\lambda_A, \lambda_D \in \mathbb{Q} \cap (1, \infty)$、一个 $\lambda_A$-非确定性折扣自动机 $\mathcal{A} = \langle \Sigma, Q_\mathcal{A}, \iota_\mathcal{A}, \delta_\mathcal{A}, \gamma_\mathcal{A} \rangle$ 以及一个 $\lambda_D$-确定性折扣自动机 $\mathcal{D} = \langle \Sigma, Q_\mathcal{D}, \iota_\mathcal{D}, \delta_\mathcal{D}, \gamma_\mathcal{D} \rangle$。当$\lambda_A = \lambda_D$时，我们可以在 $\mathcal{A}$ 和 $\mathcal{D}$的乘积上构造一个 $\lambda_A$-非确定性折扣自动机 $\mathcal{C} \equiv \mathcal{A} - \mathcal{D}$，并计算 lowword($\mathcal{C}$)。

当 $\lambda_A \neq \lambda_D$，**我们首先考虑** $\lambda_A < \lambda_D$ 的情况。

我们的算法展开 $\mathcal{A}$ 和 $\mathcal{D}$的计算树，直到某一层级；在该层级，只有 $\mathcal{A}$ 的最小值后缀词仍然相关——由于与 $\mathcal{D}$中的累积折扣因子相比，$\mathcal{A}$ 中的累积折扣因子差异巨大，因此，在 $\mathcal{A}$中不继续采用最小值后缀词所造成的任何“惩罚”（下文定义为 $m_\mathcal{A}$），即使由 $\mathcal{D}$的最大值词带来的“收益”也无法弥补；该收益至多为 maxdiff($\mathcal{D}$)。因此，在该层级，只需在 $\mathcal{A}$ 的最小值后缀中寻找能使 $\mathcal{D}$的值最高的那个。

对于每个转移 $t = q\ \sigma\ q' \in \delta_\mathcal{A}$ $\mathcal{A}^q$ $q\ \sigma\ q' = \gamma_\mathcal{A}\ q\ \sigma\ q'$ $\mathcal{A}^q$ $t$ $+(,,)$，令 $\frac{1}{\lambda_A} \cdot$ minval$(,,)$ $(,,)$ $\mathcal{A}^{q'}$ lowword$(\ )$ 表示以 作为第一步转移所能得到的最佳（最小）值

。如果 $t$ 是优选转移，即它是 $\mathcal{A}^q$的一条最小值无限运行的起始转移，我们就称之为优选转移，具体而言， $\delta_{pr} = \{ t = (q, \sigma, q') \in \delta_{\mathcal{A}} \mid \text{minval}(t) = \text{lowword}(\mathcal{A}^q) \}$ 是 $\mathcal{A}$的优选转移集合。注意， $\mathcal{A}^q$ 中一条仅经过 $\delta_{pr}$中的转移的无限运行，其值等于 $\text{lowrun}(\mathcal{A}^q)$（参见 [10, 定理的证明 9]）。

如果 $\mathcal{A}$ 的所有转移都是优选转移，那么 $\mathcal{A}$ 在所有词上的值都相同，因此 $\min\{\mathcal{A}(w) - \mathcal{D}(w) \mid w \in \Sigma^\omega\} = \text{lowrun}(\mathcal{A}) - \text{highword}(\mathcal{D})$。（回想一下，由于 $\mathcal{D}$ 是确定性的，我们可以轻松计算 $\text{highword}(\mathcal{D})$。）否则，令$m_{\mathcal{A}}$ 为在 $\mathcal{A}$中不采取优选转移所受到的最小惩罚，即$m_{\mathcal{A}} = \min \left\{ \text{minval}(t') - \text{minval}(t'') \;\middle|\; t' = (q, \sigma', q') \in \delta_{\mathcal{A}} \setminus \delta_{pr}, t'' = q\ \sigma''\ q'' \quad (,,) \in \delta_{pr} \right\}$。注意到$m_{\mathcal{A}} > 0$。

考虑到 $m_{\mathcal{A}}$ 与 $\text{maxdiff}(\mathcal{D})$之间的联系，首先注意，如果 $\text{maxdiff}(\mathcal{D}) = 0$, 那么 $\mathcal{D}$ 在所有词上的值都相同，因此我们有 $\min\{\mathcal{A}(w) - \mathcal{D}(w) \mid w \in \Sigma^\omega\} = \text{lowrun}(\mathcal{A}) - \text{lowrun}(\mathcal{D})$。否则，也就是说，$\text{maxdiff}(\mathcal{D}) > 0$，我们展开 $\mathcal{A}$ 和 $\mathcal{D}$ 的计算树，直到前$k$ 个层级，使得 $\mathcal{D}$中后缀运行之间的最大差值除以 $\mathcal{D}$ 的累积折扣因子后，小于在 $\mathcal{A}$中不采取优选转移所受到的最小惩罚除以 $\mathcal{A}$的累积折扣因子。也就是说， $k$ 是满足以下条件的最小整数

$$\frac{\text{MAXDIFF}(\mathcal{D})}{{\lambda_D}^k} < \frac{m_{\mathcal{A}}}{{\lambda_A}^k} \tag{3}$$

从层级 $k$开始，执行 $\mathcal{A}$的非优选转移所产生的惩罚，无法由 $\mathcal{D}$的更高值词来补偿。

在层级 $k$，我们分别考虑 $\psi$ 在某个前缀词 $u$上的每个运行 $\mathcal{A}$ 。我们应寻找一个后缀词 $w$，使其最小化

$$\mathcal{A}(uw) - \mathcal{D}(uw) = \mathcal{A}(\psi) + \frac{1}{{\lambda_A}^k} \cdot \mathcal{A}^{\delta_{\mathcal{A}}(\psi)}(w) - \mathcal{D}(u) - \frac{1}{{\lambda_D}^k} \cdot \mathcal{D}^{\delta_{\mathcal{D}}(u)}(w) \tag{4}$$

算法的一个核心要点是，使 $\mathcal{A} - \mathcal{D}$最小的每个词，都必须从层级 $k$ 开始仅执行 $\mathcal{A}$ 的优选转移（见引理 4）。由于层级 $k$ 之后所有可能的延续在 $\mathcal{A}$中产生相同的值，因此我们可以从中选择在 $\mathcal{D}$中产生最高值的延续。

令 $\mathcal{B}$ 为具有 $\mathcal{A}$的状态、但仅包含其优选转移 $\delta_{pr}$的部分自动机。（我们忽略 $\mathcal{B}$ 没有运行的词。）我们将使用自动机乘积 $\mathcal{B}^{\delta_{\mathcal{A}}(\psi)} \times \mathcal{D}^{\delta_{\mathcal{D}}(u)}$，迫使后缀词仅执行 $\mathcal{A}$的优选转移，同时计算其中在 $\mathcal{D}$中的最高值。

令 $\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))} = \langle \Sigma、\ Q_{\mathcal{A}} \times Q_{\mathcal{D}}、\ \{ (\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u)) \}、\ \delta_{pr} \times \delta_{\mathcal{D}}、\ \gamma_{\mathcal{C}} \rangle$为由 $\mathcal{B}^{\delta_{\mathcal{A}}(\psi)}$ 与 $\mathcal{D}^{\delta_{\mathcal{D}}(u)}$的乘积生成的部分 $\lambda_D$-非确定性折扣自动机，同时仅考虑 $\mathcal{D}$ 的权重（及折扣因子），即 $\gamma_{\mathcal{C}}((q,p)\ \sigma,\ (q',p')) = \gamma_{\mathcal{D}}(p, \sigma, p')$。

词 $w$ 在 $\mathcal{A}^{\delta_{\mathcal{A}}(\psi)}$ 中存在一条仅使用优选转移的运行，当且仅当 $w$ 在 $\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))}$中存在一条运行。此外，注意到 $\mathcal{C}$ 中的非确定性仅与 $\mathcal{A}$中的非确定性有关，而 $\mathcal{C}$ 的权重函数仅取决于

$\mathcal{D}$的权重，因此，$\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))}$在同一个词上的所有运行都会得到相同的值，该值就是该词在 $\mathcal{D}$中的值。结合这两点，我们得到：词 $w$ 在 $\mathcal{A}^{\delta_{\mathcal{A}}(\psi)}$中存在一条仅使用优选转移的运行，当且仅当$w$ 在 $\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))}$中存在一条运行 $r$，使得 $\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))}(r) = \mathcal{D}^{\delta_{\mathcal{D}}(u)}(w)$。因此，在取出 $\mathcal{A}$的长度为 $k$的运行 $\psi$ 后，按照式（4）中定义的记号，一个只能使用 $\mathcal{A}$的优选转移、且使$\mathcal{D}^{\delta_{\mathcal{D}}(u)}(w)$最大的后缀词 $w$，其值为 $\mathcal{D}^{\delta_{\mathcal{D}}(u)}(w) =\text{highrun}(\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))})$。由此可得

$$\min\{\mathcal{A}(v)-\mathcal{D}(v)\mid v\in\Sigma^{\omega}\}=$$
$$\min\left\{\mathcal{A}(\psi)+\frac{\mathcal{A}^{\delta_{\mathcal{A}}(\psi)}(w)}{{\lambda_A}^k}-\mathcal{D}(u)-\frac{\mathcal{D}^{\delta_{\mathcal{D}}(u)}(w)}{{\lambda_D}^k}\middle|\begin{array}{c}u\in\Sigma^k,w\in\Sigma^{\omega},\\ \psi\text{ is a run of }\mathcal{A}\text{ on }u\end{array}\right\}=$$
$$\min_{\psi}\left\{\mathcal{A}(\psi)+\frac{\text{LOWRUN}(\mathcal{A}^{\delta_{\mathcal{A}}(\psi)})}{{\lambda_A}^k}-\mathcal{D}(u)-\frac{\text{HIGHRUN}(\mathcal{C}^{(\delta_{\mathcal{A}}(\psi),\delta_{\mathcal{D}}(u))})}{{\lambda_D}^k}\middle|\begin{array}{c}u\in\Sigma^k,\\ \psi\text{ is a run}\\ \text{of }\mathcal{A}\text{ on }u\end{array}\right\}$$

因此，只需针对 $\mathcal{A}$ 的每个大小为 $k$ 的运行计算此值，也就是说，针对 $\mathcal{A}$ 的计算树中的每个叶节点计算此值。

**的情况**与此类似，只需做出以下更改：

– 对于 $\mathcal{D}$ 的每个转移，我们计算 $\text{maxval}(p,\sigma,p') = \gamma_{\mathcal{D}}(p,\sigma,p') + \frac{1}{\lambda_D}\cdot \text{highword}(\mathcal{D}^{p'})$，而不是 $\text{minval}(q,\sigma,q')$。– $\mathcal{D}$ 的优选转移是那些以值最大的无限运行开头的转移，即 $\delta_{pr}=\{t=(p,\sigma',p')\in\delta_{\mathcal{D}}\mid \text{maxval}(t)=\text{highrun}(\mathcal{D}^{p})\}$；最小惩罚 $m_{\mathcal{D}}$ 为 $m_{\mathcal{D}}=\min\left\{\text{maxval}(t'')-\text{maxval}(t')\ \middle|\ \begin{array}{c}t''=(p,\sigma'',p'')\\ \in\delta_{pr}\end{array},t'=\ p\ \sigma'\ p'\ \in\delta_{\mathcal{D}}\setminus\delta_{pr}\right\}$(, , )。– $k$ 应为满足 $\text{maxdiff}_{{\lambda_A}^k}(\mathcal{A})<\frac{m_{\mathcal{D}}}{{\lambda_D}^k}$ 的最小整数。– 我们将 $\mathcal{B}$ 定义为 $\mathcal{D}$ 在其优选转移上的限制，并将 $\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))}$ 定义为乘积 $\mathcal{A}^{\delta_{\mathcal{A}}(\psi)}$ 和 $\mathcal{B}^{\delta_{\mathcal{D}}(u)}$ 上的一个部分 $\lambda_A$- 非确定性折扣自动机，同时考虑 $\mathcal{A}$ 的权重。然后，我们针对 $\mathcal{A}$, $\psi$ 的每个大小为 $k$ 的运行计算 $\text{lowrun}(\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))})$，并得出 $\min\{\mathcal{A}-\mathcal{D}\}$ 等于 $\min_{\psi}\{\mathcal{A}(\psi)+\text{lowrun}(\mathcal{C}^{(\delta_{\mathcal{A}}(\psi),\delta_{\mathcal{D}}(u))})_{{\lambda_A}^k}-\mathcal{D}(u)-\text{highrun}(\mathcal{D}_{\delta_{\mathcal{D}}(u)})_{{\lambda_D}^k}\}$。

请注意，在这种情况下，同一个词上的 $\mathcal{C}^{(\delta_{\mathcal{A}}(\psi)\ \delta_{\mathcal{D}}(u))}$ 的所有运行可能不具有相同的值，但这并非必要，因为我们要寻找最小运行值（即最小词值）。

□

请注意，如果反转包含关系的方向，即考虑一个确定性 $\mathcal{A}$ 和一个非确定性$\mathcal{D}$，引理 3 中的算法将无法奏效。要找出 $\mathcal{B}^{\delta_{\mathcal{A}}(\psi)}\times\mathcal{D}^{\delta_{\mathcal{D}}(u)}$中有效词的最大值，需要 $\mathcal{D}$ 具有确定性。如果 $\mathcal{D}$ 不是确定性的，那么在某个词 $w$ 上，$\mathcal{B}^{\delta_{\mathcal{A}}(\psi)}\times\mathcal{D}^{\delta_{\mathcal{D}}(u)}$ 的最大值运行的值等于 $\mathcal{D}$ 在 $w$上的某个运行的值，但不一定等于 $\mathcal{D}$ 在 $w$上的值。此外，在 $\lambda_A>\lambda_D$的情况下，计算 $\text{highword}(\mathcal{D}^{p})$ 也需要 $\mathcal{D}$ 具有确定性。

为证明引理 3 的正确性，我们提出以下断言。

**引理 4.** 对于每组输入折扣因子 $\lambda_A, \lambda_D \in \mathbb{Q} \cap (1, \infty)$，只要$\lambda_A < \lambda_D$, $\lambda_A$-非确定性折扣自动机 $\mathcal{A}$ 和 $\lambda_D$-确定性折扣自动机 $\mathcal{D}$成立，那么任何使$\mathcal{A}(w) - \mathcal{D}(w)$ 达到最小值的无限字 $w$，在$\text{MAXDIFF}_{{\lambda_D}^k}(\mathcal{D}) < \frac{m_\mathcal{A}}{{\lambda_A}^k}$ 成立的每个层级 $k$ 上，都必须采取 $\mathcal{A}$ 的优选转移。

证明。 考虑折扣因子 $\lambda_A, \lambda_D \in \mathbb{Q} \cap (1, \infty)$，使得 $\lambda_A < \lambda_D$, $\lambda_A$-非确定性折扣自动机 $\mathcal{A}$, $\lambda_D$-确定性折扣自动机 $\mathcal{D}$，且 $k$ 是满足下述条件的最小整数

$$\frac{\text{MAXDIFF}(\mathcal{D})}{{\lambda_D}^k} < \frac{m_\mathcal{A}}{{\lambda_A}^k}$$

反设存在一个词 $v$，使其将 $\mathcal{A}-\mathcal{D}$ 最小化，但 $\psi_\mathcal{A}$ 上 $\mathcal{A}$ 对 $v$ 的一个取最小值的运行，在某个层级 $n \geq k$ 未进行优选转移。令 $u$ 为 $n$ 长的 $v$ 前缀， $w$ 为相应的后缀（即 $v = u \cdot w$）， $\psi$ 为 $\psi_\mathcal{A}$ 在 $u$ 上的前缀运行，且 $w'$ 为 $\mathcal{A}^{\delta_\mathcal{A}(\psi)}$ 的某个取最小值的词。当 $\psi_\mathcal{A}$ 继续处理 $w$ 时所进行的第一次转移不是优选转移，即

$$\mathcal{A}^{\delta_\mathcal{A}(\psi)}(w) \geq \text{LOWWORD}(\mathcal{A}^{\delta_\mathcal{A}(\psi)}) + m_\mathcal{A} = \mathcal{A}^{\delta_\mathcal{A}(\psi)}(w') + m_\mathcal{A} \tag{5}$$

因此,

$$\begin{aligned}
\mathcal{A}(v) - \mathcal{D}(v) &\overset{(4)}{=} \mathcal{A}(\psi) + \frac{\mathcal{A}^{\delta_\mathcal{A}(\psi)}(w)}{{\lambda_A}^n} - \mathcal{D}(u) - \frac{\mathcal{D}^{\delta_\mathcal{D}(u)}(w)}{{\lambda_D}^n} \\
&\overset{(5),(2)}{\geq} \mathcal{A}(\psi) + \frac{\mathcal{A}^{\delta_\mathcal{A}(\psi)}(w') + m_\mathcal{A}}{{\lambda_A}^n} - \mathcal{D}(u) - \frac{\text{LOWRUN}(\mathcal{D}^{\delta_\mathcal{D}(u)}) + \text{MAXDIFF}(\mathcal{D})}{{\lambda_D}^n} \\
&\overset{(3)}{>} \mathcal{A}(\psi) + \frac{\mathcal{A}^{\delta_\mathcal{A}(\psi)}(w')}{{\lambda_A}^n} - \mathcal{D}(u) - \frac{\text{LOWRUN}(\mathcal{D}^{\delta_\mathcal{D}(u)})}{{\lambda_D}^n} \\
&\overset{(2)}{\geq} \mathcal{A}(\psi) + \frac{\mathcal{A}^{\delta_\mathcal{A}(\psi)}(w')}{{\lambda_A}^n} - \mathcal{D}(u) - \frac{\mathcal{D}^{\delta_\mathcal{D}(u)}(w')}{{\lambda_D}^n} \\
&\overset{(4)}{=} \mathcal{A}(u \cdot w') - \mathcal{D}(u \cdot w')
\end{aligned}$$

从而导出矛盾。 □

转而考虑有限词上的自动机，我们通过在字母表中添加一个表示词结束的新字母，并进行一些必要的调整，将问题归约为引理 3 中处理的相应问题。

**引理 5.** 存在一种算法，对于任意输入折扣因子$\lambda_A$， $\lambda_D \in \mathbb{Q} \cap (1, \infty)$， $\lambda_A$-非确定性折扣自动机 $\mathcal{A}$和 $\lambda_D$-确定性折扣自动机 $\mathcal{D}$，可计算它们在有限词上的值下确界 $\{\, \mathcal{A}(u) - \mathcal{D}(u) \mid u \in \Sigma^+ \,\}$，并判断是否存在一个有限词 $u$，使得 $\mathcal{A}(u) - \mathcal{D}(u)$ 等于该值。

证明。 不失一般性，我们假设自动机的初始状态没有入转移。（每个自动机都可以在线性时间内转换为具有此性质的等价自动机。）

我们按如下方式，将一个处理有限词的非确定性折扣自动机 $\mathcal{N}$ 转换为一个处理无限词的非确定性折扣自动机 $\hat{\mathcal{N}}$，使得 $\hat{\mathcal{N}}$ 直观上模拟 $\mathcal{N}$ 的有限运行。对于字母表 $\Sigma$、折扣因子 $\lambda \in \mathbb{Q} \cap (1,\infty)$，以及一个处理有限词的 $\lambda$-非确定性折扣自动机（确定性折扣自动机）$\mathcal{N} = \langle \Sigma, Q_{\mathcal{N}}, \iota_{\mathcal{N}}, \delta_{\mathcal{N}}, \gamma_{\mathcal{N}} \rangle$，我们定义处理无限词的 $\lambda$-非确定性折扣自动机（确定性折扣自动机）$\hat{\mathcal{N}} = \hat{\langle} \Sigma, Q_{\mathcal{N}} \cup \{q_\tau\}, \iota_{\mathcal{N}}, \hat{\delta}_{\mathcal{N}}, \hat{\gamma}_{\mathcal{N}} \rangle$。新字母表 $\Sigma = \Sigma \cup \{\tau\}$ 包含一个新字母 $\tau \notin \Sigma$，用于标记有限词的结束。新状态 $q_\tau$ 在字母表中的每个字母上都有值为 0 的自环，并且每个非初始状态都有一条在新字母 $\tau$ 上、指向 $q_\tau$ 且值为 0 的转移。形式化地，$\hat{\delta}_{\mathcal{N}} = \delta_{\mathcal{N}} \cup \{ (q_\tau, \sigma, q_\tau) \mid \sigma \in \hat{\Sigma} \} \cup \{ (q, \tau, q_\tau) \mid q \in Q_{\mathcal{N}} \setminus \iota_{\mathcal{N}} \}$, and

$$\gamma_{\hat{\mathcal{N}}}(t) = \begin{cases} \gamma_{\mathcal{N}}(t) & t \in \delta_{\mathcal{N}} \\ 0 & \text{otherwise} \end{cases}$$

Ob在每个状态 $q \in Q_{\mathcal{N}}$ 处服务 t，以下条件成立。

1. 对于 $\mathcal{N}^q$ 的每个有限运行 $r_{\mathcal{N}}$，存在 $\mathcal{N}^q$ 的一个无限运行 $r \hat{}_{\mathcal{N}}$，使得 $\hat{\mathcal{N}}^q(r \hat{}_{\mathcal{N}}) = \mathcal{N}^q(r_{\mathcal{N}})$，且 $r \hat{}_{\mathcal{N}}$ 进行某些 $\tau$ 转移。（$r \hat{}_{\mathcal{N}}$ 可以从 $r_{\mathcal{N}}$ 开始，然后仅通过 $\tau$ 转移继续。）2. 对于具有一个 $\tau$ 转移的 $\hat{\mathcal{N}}^q$ 的每个无限运行 $r \hat{}_{\mathcal{N}}$，存在 $\mathcal{N}^q$ 的一个有限运行 $r_{\mathcal{N}}$，使得 $\hat{\mathcal{N}}^q(r \hat{}_{\mathcal{N}}) = \mathcal{N}^q(r_{\mathcal{N}})$。（$r_{\mathcal{N}}$ 可以是 $r \hat{}_{\mathcal{N}}$ 在第一次 $\tau$ 转移之前的最长前缀。）3. 对于不具有 $\tau$ 转移的 $\hat{\mathcal{N}}^q$ 的每个无限运行 $r \hat{}_{\mathcal{N}}$，存在由 $\mathcal{N}^q$ 组成的有限运行级数，使得 $\mathcal{N}^q$ 中这些运行的值收敛于 $\hat{\mathcal{N}}^q(r \hat{}_{\mathcal{N}})$。（例如，$r \hat{}_{\mathcal{N}}$ 的所有前缀组成的级数。）

因此，对于每个 $q \in Q_{\mathcal{N}}$，$\inf \{ \mathcal{N}^q(r) \mid r \text{ 都是 } \mathcal{N}^q \} = \text{lowrun}(\hat{\mathcal{N}}^q)$ 的一次运行，而 $\sup \{ \mathcal{N}^q(r) \mid r \text{ 是 } \mathcal{N}^q \} = \text{highrun}(\hat{\mathcal{N}}^q)$ 的一次运行。（对于非初始状态 $q$，我们也考虑 $\mathcal{N}^q$ 在空词上的“运行”，并将其值定义为 0。）请注意，当且仅当存在一条 $\hat{\mathcal{N}}^q$ 的无限运行取得该值且执行了一次 $\tau$ 转移时，$\mathcal{N}^q$ 的下确界（上确界）运行值才由 $\mathcal{N}^q$ 的一次实际运行取得。

对于每个状态 $q \in Q_{\mathcal{N}}$，我们可以计算 $\text{lowrun}(\hat{\mathcal{N}}^q)$ 是否由一条执行 $\tau$ 转移的运行取得。我们可以按如下方式计算所有状态的 $\text{lowrun}(\hat{\mathcal{N}}^q)$，然后启动一个过程，迭代地标记 $\hat{\mathcal{N}}$ 的状态，使得最终，当且仅当存在一条执行 $\tau$ 转移的运行能够取得 $\text{lowrun}(\hat{\mathcal{N}}^q)$ 时，$q \in Q_{\hat{\mathcal{N}}}$ 才会被标记。我们首先只标记 $q_\tau$。在每次迭代中，我们进一步标记每个状态 $q$：从该状态出发，存在一条优选转移 $t = (q, \sigma, q') \in \delta_{pr}$，转移到某个已标记状态 $q'$。当某次迭代没有新状态需要标记时，该过程便终止。类似地，我们可以确定 $\text{highrun}(\hat{\mathcal{N}}^q)$ 是否由一条到达 $q_\tau$ 的运行取得。

考虑折扣因子 $\lambda_A, \lambda_D \in \mathbb{Q} \cap (1,\infty)$、有限词上的 $\lambda_A$-非确定性折扣自动机 $\mathcal{A}$ 和 $\lambda_D$-确定性折扣自动机 $\mathcal{D}$。当 $\lambda_A = \lambda_D$ 时，与引理 3 类似，算法使用 $\mathcal{C}$ 求出 $\mathcal{C} \equiv \mathcal{A} - \mathcal{D}$ 的下确界值，并使用上述过程判断是否存在某个有限词能够达到该值。

否则，并假设 $\lambda_A < \lambda_D$。我们将引理 3 中对 $\mathcal{A}$ 转换为 $\mathcal{D}$ 的方式进行。根据上述观察，我们有 $\inf \{ \mathcal{A}(u) - \mathcal{D}(u) \mid u \in \Sigma^+ \} = \text{最小值}\{ \hat{\mathcal{A}}(w) - \hat{\mathcal{D}}(w) \mid w \in \Sigma^\omega \}$，并且

inf $\{\mathcal{A}(u) - \mathcal{D}(u)\}$ 可达，当且仅当存在某个包含 $\tau$ 转移的词，使 $\min\{\hat{\mathcal{A}}(w) - \hat{\mathcal{D}}(w)\}$ 可达。因此，在计算 lowrun 或 highrun 时，我们也执行上述过程，以确定该值是否可由包含 $\tau$ 转移的运行达到。我们判定，inf $\{\mathcal{A}(u) - \mathcal{D}(u)\}$ 可达，当且仅当计算树中存在一片叶子可以导向该值，且相关的 lowrun 和 highrun 值均可达。□

**复杂性分析** 我们将在下文证明，引理 3 和引理 5 中的算法只需使用相对于输入自动机规模为多项式的空间，因此相应判定问题可由多项式空间算法解决。我们将非确定性折扣自动机 $\mathcal{N}$ 的规模定义为 $|\mathcal{N}|$，即其转移数量、其中任意权重的最大二进制表示长度，以及折扣因子的最大一元表示长度三者中的最大值。（折扣因子的二进制表示可能会导致我们的算法使用指数级空间，特别是当两个因子非常接近时。）输入的非确定性折扣自动机可以具有有理数权重，但考虑与之等价的整数权重非确定性折扣自动机会更方便；这类自动机通过将所有权重乘以其公分母 [6] 得到。（注意，这会使所有词的值都乘以同一比例，同时输入规模至多发生多项式变化。）

在讨论复杂性分析之前，我们先给出一个辅助引理。

**引理 6.** 对于任意整数 $p > q \in \mathbb{N}\setminus\{0\}$、一个权重为整数的 $\frac{p}{q}$-非确定性折扣自动机 $\mathcal{A}$，以及其套索运行 $r = t_0, t_1, \ldots, t_{x-1}, (t_x, t_{x+1}, \ldots, t_{x+y-1})^\omega$，存在一个整数 $b$，使得 $\mathcal{A}(r) = \frac{b}{(p^y - q^y)p^x}$。

证明。设 $\lambda = \frac{p}{q}$ 为 $\mathcal{A}$的折扣因子，$\gamma$ 为其权重函数。考虑 $\mathcal{A}$的套索运行 $r = t_0, t_1, \ldots, t_{x-1}, (t_x, t_{x+1}, \ldots, t_{x+y-1})^\omega$。设 $v_f = \gamma(t_0) + \frac{1}{\lambda}\gamma(t_1) + \ldots + \frac{1}{\lambda^{x-1}}\gamma(t_{x-1})$ 为其前缀值，且 $v_\ell = \gamma(t_x) + \frac{1}{\lambda}\gamma(t_{x+1}) + \ldots + \frac{1}{\lambda^{y-1}}\gamma(t_{x+y-1})$ 为其环路值。

由于所有权重都是整数，因此对于某些整数 $a_f$ 和 $a_\ell$，有 $v_f = \frac{a_f}{p^x}$ 和 $v_\ell = \frac{a_\ell}{p^y}$。回想一下，在一个 $\lambda$-非确定性折扣自动机中，长度为 $y$、累积值为 $v_\ell$ 的环路 $\ell$，其无限重复的累积值为 $\sum_{i=0}^{\infty} \frac{v_\ell}{\lambda^{y(\ell)i}} = v_\ell \frac{\lambda^y}{\lambda^y - 1}$。因此，$r$ 的值为

$$\gamma(r) = v_f + \frac{1}{\lambda^x} \cdot v_\ell \frac{\lambda^y}{\lambda^y - 1} = \frac{a_f}{p^x} + \frac{a_\ell}{p^y} \cdot \frac{1}{\lambda^{x-y}(\lambda^y - 1)} = \frac{a_f}{p^x} + \frac{a_\ell \cdot q^{x-y}}{p^{y+x-y}(\frac{p^y - q^y}{q^y})}$$
$$= \frac{a_f(p^y - q^y) + a_\ell \cdot q^x}{p^x(p^y - q^y)}$$

□

接下来进行复杂性分析。设输入规模为 $S = |\mathcal{A}| + |\mathcal{D}|$，$\lambda_\mathcal{A}$ 和 $\lambda_\mathcal{D}$ 的约化形式分别为 $\frac{p}{q}$ 和 $\frac{p_\mathcal{D}}{q_\mathcal{D}}$，$\mathcal{A}$ 中的状态数为 $n$，且 $\mathcal{D}$ 中转移权重的最大差值为 $M$。注意

对于 $n \le S$、$p \le S$、$M \le 2 \cdot 2^S$、$\frac{\lambda_{\mathcal{D}}}{\lambda_D - 1} \le \frac{p_{\mathcal{D}}}{p_{\mathcal{D}} - q_{\mathcal{D}}} \le p_{\mathcal{D}} \le S$，以及 $\lambda_{\mathcal{D}} > \lambda_{\mathcal{A}} > 1$，我们还有 $\frac{\lambda_{\mathcal{D}}}{\lambda_{\mathcal{A}}} = \frac{p \cdot q_{\mathcal{D}}}{q \cdot p_{\mathcal{D}}} \ge 1 + \frac{1}{S^2}$。

注意，$\mathcal{A}$ 有一个最佳无限运行（而 $\mathcal{D}$ 有一个最差无限运行），其形式如引理 6 中所示的套索形式，带有 $x, y \in [1..n]$。事实上，沿着优选转移，运行必定会完成一个套索，随后可以永远重复其优选转移的选择。因此，$m_{\mathcal{A}}$作为两个套索运行之差，其形式为

$$m_{\mathcal{A}} = \frac{b_1}{p^{x_1}(p^{y_1} - q^{y_1})} - \frac{b_2}{p^{x_2}(p^{y_2} - q^{y_2})} = \frac{b_3}{p^n(p^{y_1} - q^{y_1})(p^{y_2} - q^{y_2})} > \frac{b_3}{p^n p^{y_1} p^{y_2}}$$
$$\ge \frac{1}{p^{3n}} \ge \frac{1}{S^{3S}} \overset{\text{for } S \ge 1}{>} \frac{1}{(2^S)^{3S}} = \frac{1}{2^{3S^2}}$$

对于某些 $x_1, x_2, y_1, y_2 \le n$，以及某些整数 $b_1, b_2, b_3$。（类似地，我们可以证明 $m_{\mathcal{D}} > \frac{1}{2^{3S^2}}$。）我们有 maxdiff($\mathcal{D}$) $\le M \cdot \frac{\lambda_{\mathcal{D}}}{\lambda_{\mathcal{D}} - 1}$，因此

$$\frac{\text{MAXDIFF}(\mathcal{D})}{m_{\mathcal{A}}} \le \frac{M \cdot \frac{\lambda_{\mathcal{D}}}{\lambda_{\mathcal{D}} - 1}}{m_{\mathcal{A}}} \le \frac{2^{1+S} \cdot S}{m_{\mathcal{A}}} \overset{\text{(for } S \ge 1)}{<} \frac{2^{3S}}{m_{\mathcal{A}}} < 2^{3S+3S^2}$$

回想一下，我们将计算树展开到层级 $k$，这是满足 $(\frac{\lambda_D}{\lambda_A})^k > \text{maxdiff}_{m_{\mathcal{A}}}(\mathcal{D})$ 的最小整数。注意，对于 ~~$S \ge 1$~~，我们有 $\left(\frac{\lambda_{\mathcal{D}}}{\lambda_{\mathcal{A}}}\right)^{S^2} \ge \left(1 + \frac{1}{S^2}\right)^{S^2} \ge 2$，因此对于 $k' = S^2 \cdot (3S + 3S^2)$，我们有

$$\left(\frac{\lambda_D}{\lambda_A}\right)^{k'} = \left(\left(\frac{\lambda_D}{\lambda_A}\right)^{S^2}\right)^{3S+3S^2} \ge 2^{3S+3S^2} > \frac{\text{MAXDIFF}(\mathcal{D})}{m_{\mathcal{A}}}$$

这意味着 $k$ 是关于 $S$的多项式。类似的分析表明，对于 $\lambda_{\mathcal{D}} < \lambda_{\mathcal{A}}$，$k$ 也是关于 $S$ 的多项式。

考虑使用我们算法的判定问题，由于 NPSPACE 与 PSPACE 等价，该算法可以非确定性地逐个字母猜测一个大小为 $k$的最优前缀词 $u$，以及 $u$上的一个运行 $\psi$，并逐个转移地计算 $\mathcal{A}(\psi)$+lowrun$_{(\mathcal{A}^{\delta_{\mathcal{A}}(\psi)}\ \lambda_A{}^k}$ $\overline{\mathcal{D}(u)} -$ highrun$_{(\mathcal{C}^{(\delta_{\mathcal{A}}(\psi), \delta_{\mathcal{D}}(u))})\ \lambda_D{}^k}$ 的值。

请注意，在算法的运行过程中，我们需要保存以下信息，这可以在多项式空间内完成：

– 自动机 $\mathcal{C} \equiv \mathcal{B} \times \mathcal{D}$（或 $\mathcal{A} \times \mathcal{B}$），这需要多项式空间。– $\lambda_{\mathcal{A}}{}^k$（用于 $\mathcal{A}(\psi)$）和 $\lambda_{\mathcal{D}}{}^k$（用于 $\mathcal{D}(u)$）。由于我们以二进制表示法保存它们，因此有 $\log_2(\lambda^k) \le k \log_2(S)$，需要多项式空间。

因此，我们得到以下复杂度结果。

**定理 4.** 对于输入折扣因子 $\lambda_A, \lambda_D \in \mathbb{Q} \cap (1, \infty)$，在有限字或无限字上运行的 $\lambda_A$-非确定性折扣自动机 $\mathcal{A}$和$\lambda_D$-确定性折扣自动机 $\mathcal{D}$，判断$\mathcal{A}(w) \ge \mathcal{D}(w)$ 以及判断 $\mathcal{A}(w) > \mathcal{D}(w)$ 是否对所有词 $w$成立，均可在多项式空间内完成。

证明。 对于无限字的情形，我们使用引理 3；对于有限词的情形，我们使用引理 5，检查 $\min\{\mathcal{A}(w) - \mathcal{D}(w)\} < 0$是否成立，以及 $\min\{\mathcal{A}(w) - \mathcal{D}(w)\} \leq 0$是否成立。对于有限词的情形，我们还会利用是否存在一个实际的词能取得所需值这一信息。 □

由于整数折扣非确定性自动机总是可以确定化的 [8], ，我们可以推论出，存在一种算法可以判定具有不同（或相同）折扣因子的整数折扣非确定性自动机之间的等价性，以及严格包含和非严格包含关系。不过请注意，该问题可能不属于多项式空间，因为确定化会使状态数量呈指数级增长，导致 $k$ 在 $S$ 中呈指数级增长；而以二进制表示存储数量级为 $\lambda^k$ 的值，可能需要指数级空间。

**推论 1.** 对于输入的整数折扣因子$\lambda_A$， $\lambda_B \in \mathbb{N}$, $\lambda_A$-非确定性折扣自动机 $\mathcal{A}$ 和 $\lambda_B$-非确定性折扣自动机 $\mathcal{B}$ ，存在算法可以判定对于有限词或无限词，是否对所有词 $w$均有 $\mathcal{A}(w) > \mathcal{B}(w)$、 $\mathcal{A}(w) \geq \mathcal{B}(w)$，或 $\mathcal{A}(w) = \mathcal{B}(w)$ 。

## 5 结论

这一新的可判定性结果为比较具有不同整数折扣因子的折扣和自动机提供了一种算法，这可能有助于拓展折扣和自动机在形式化验证中的应用；而不可判定性结果则进一步说明，将具有多个整数折扣因子的折扣和自动机限制为整洁的非确定性多折扣自动机是合理的。该新算法也拓展了折扣因子为有理数的折扣和自动机原本较有限的应用范围，而这一方向仍值得进一步研究。

**致谢** 感谢 Guillermo A. Perez 与我们就不同折扣因子的整数折扣非确定性自动机之间的比较进行了富有启发性的讨论。

## 参考文献

1. de Alfaro, L., Henzinger, T.A., Majumdar, R.：系统理论中的未来折扣。载于：ICALP 论文集。第 2719 卷，第 1022–1037 页（2003年）。https://doi.org/ 10 1007 3 540 45061 0_79 ./---. Almagor, S., Boker, U., Kupfer 2 man, O.：加权自动机有哪些问题是可判定的? Information and Computatio **282** （ 2022年）。https://doi.org/10104651 310162020./j.ic... Almagor, S., Kupferman, O., Ringert, J.O., Velner, Y.：定量假设保证综合。载于：CAV 论文集。第 353–374 页。施普林格（2017年）。https://doi.org/ 10 1007 978 3 319 63390 9_19 ./----. 4 Andersson, D.：折扣收益博弈的改进算法。载于：ESSLLI 学生会议论文集。第 91–98 页（2006年）。5. Bansal, S., Chaudhuri, S., Vardi, M.Y.：定量验证中的比较器自动机。载于：FoSSaCS 论文集。LNCS，第 10803 卷，第 420–437 页（2018年）。https://doi.org/10.1007/978-3-319-89366-2_23

6. Boker, U., Hefetz, G.：具有多个折扣因子的折扣和自动机。载于：CSL 论文集。LIPIcs，第 183 卷，第 12:1–12:23 页。Schloss Dagstuhl - Leibniz- 信息学中心（2021年12月 7日 10 4230 2021年）。https://doi.org/ . /LIPIcs.CSL. . . Boker, U., Hefetz, G.：关于具有多个折扣因子的折扣和自动机的比较。载于：FoSSaCS 论文集。第 371–391 页（2023年）。https://doi.org/ 10 1007 978 3 031 30829 1 _18 . / - - - - . Boker, U., Henzinger, T.A.[8]：折扣和自动机的精确与近似确定化。Log. Methods Comput. Sci. 10(1)（2014年）。https://doi.org/10 2014 9 2168 10 1 10 . /LMCS- ( : ) . Boker, U., Henzinger, T.A., Otop, J.：目标折扣和问题。载于：LICS 论文集。第 750 74 10 761 2015 10 1109 2015 – 页（）。https://doi.org/ . /LICS. . . Boker, U., Lehtinen, K.：定量自动机中的历史确定性与适于博弈性。载于：FSTTCS 论文集。第 38:1–38:20 页（2021年）。https://doi.org/10 38 11 4230 2021 . /LIPIcs.FSTTCS. . . Brenguier, R., Clemente, L., Hunter, P., Pérez, G.A., Randour, M., Raskin, J.F., Sankur, O., Sassolas, M.：用于反应式综合的非零和博弈。载于：语言与自动机理论及应用。第 3–23 页。施普林格（2016年）。12. Chatterjee, K., Doyen, L., Henzinger, T.A.：交替加权自动机。载于：FCT 论文集。LNCS，第 5699 卷，第 3–13 页（2009年）。https://doi.org/ 10 1007 978 3 642 03409 1 _2 . / - - - - . Chatterjee, K., Doyen, L., Henzinger, T.A.[13]：概率加权自动机。载于：CONCUR 论文集。LNCS，第 5710 卷，第 244–258 页（2009年）。https://doi.org/ 10 1007 978 3 642 04081 8 _17 . / - - - - . Chatterjee, K., Doyen, L., Henzinger, T.A.[14]：定量语言的表达能力与闭包性质。Log. Methods Comput. Sci. 6(3)（2010年），http://arxiv.org/abs/1007 4018 15 . . Chatterjee, K., Doyen, L., Henzinger, T.A.：定量语言。ACM Trans. Comput. Log. 11 1805953 16 4 23 1 23 38 2010 10 1145 1805950（），：–：（）。https://doi.org/ . / . . Chatterjee, K., Forejt, V., Wojtczak, D.：图与马尔可夫决策过程中的多目标折扣奖励验证。载于：LPAR 论文集。LNCS，第 8312 卷，第 228 242 2013 10 1007 978 3 642 45221 5 _17 – 页（）。https://doi.org/ . / - - - - . Clarke, E.M., Draghicescu, I.A., Kurshan, R.P.[17]：展示不同类型的 $\omega$-自动机之间的语言包含与等价性的统一方法。Information Processing Letters 46, 301–308（1993年）。18. Degorre, A., Doyen, L., Gentilini, R., Raskin, J., Toruńczyk, S.：不完全信息下的能量博弈与平均收益博弈。载于：CSL 论文集。LNCS，第 6247 卷，第 260 274 2010 10 1007 978 3 642 15205 4 _22 – 页（）。https://doi.org/ . / - - - - . Droste, M., Kuske, D.[19]：斜形式幂级数与无穷形式幂级数。Theor. Comput. Sci. 366 024 20 3 199 227 2006 10 1016 2006 08（），–（）。https://doi.org/ . / j.tcs. . . . Filiot, E., Gentilini, R., Raskin, J.：有限值加权自动机。载于：FSTTCS 论文集。LIPIcs，第 29 卷，第 133–145 页（2014年）。https://doi.org/10 133 21 4230 2014 . / LIPIcs.FSTTCS. . . Filiot, E., Gentilini, R., Raskin, J.：由函数自动机定义的定量语言。Log. Methods Comput. Sci. 11(3)（2015年）。https://doi.org/10 2015 22 2168 11 3 14 . / LMCS- ( : ) . Filiot, E., Löding, C., Winter, S.：从有限词上定义域为部分定义的加权规约进行综合。载于：FSTTCS 论文集。第 46:1–46:16 页（2020年）。https://doi.org/10 46 23 4230 2020 . /LIPIcs.FSTTCS. . . Gimbert, H., Zielonka, W.：多重折扣马尔可夫决策过程的极限。载于：LICS 论文集。第 89–98 页（2007年）。https://doi.org/10.1109/LICS.2007.28

24. Glendinning, P., Sidorov, N.：非整数底数下实数的唯一表示。《数学研究快报》**8**(4), 535–543 (2001) 25. Hare, K.：Pisot数和Salem数的β展开。载于：滑铁卢计算机代数研讨会（2006年）26. Hojati, R., Touati, H., Kurshan, R., Brayton, R.：高效 ω-正则语言包含。载于：CAV会议论文集。LNCS，第663卷。施普林格（1992）27. Hunter, P., Pérez, G.A., Raskin, J.：无悔的反应式综合。《Acta Informatica》**54**(1), 3–39 (2017)。https://doi.org/10.1007/s00236-016-0268-z 28. Kupferman, O., Vardi, M., Wolper, P.：分支时态模型检测的自动机理论方法。《美国计算机学会会刊》**47**(2), 312–360 (2000) 29. Madani, O., Thorup, M., Zwick, U.：折扣确定性马尔可夫决策过程与折扣全点对最短路径。《ACM算法汇刊》**6**(2), 331721849 3013325 201010114517218 37：–：（）。https://doi.org/./.. Mahler, K.：关于3 2的幂的一个未解问题。《澳大利亚数学会会刊》**8**(2), 313–321 (1968) 31. Minsky, M.L.：《计算：有限与无限机器》。普伦蒂斯-霍尔自动计算丛书，普伦蒂斯-霍尔（1967）32. Sutton, R.S., G.Barto, A.：《强化学习导论》。麻省理工学院出版社（1998551283 33105555），http://dl.acm.org/doi/book/./. Tasiran, S., Hojati, R., Brayton, R.：使用非确定性ω自动机进行语言包含。载于：CHARME会议论文集。LNCS，第987卷，第261–277页。施普林格（1995）34. Vardi, M.Y.：并发程序验证：自动机理论框架。载于：LICS会议论文集，第167–176页（1987）35. Vardi, M.Y.：线性时态逻辑的自动机理论方法。载于：Moller, F., Birtwistle, G.（编）《并发逻辑：结构与自动机》。LNCS，第1043卷，第238–266页（1996）36. Wang, Y., Ye, Q., Liu, T.：超越指数折扣和：回报函数的自动学习。CoRR（201911591 371905），http://arxiv.org/abs/.. Zwick, U., Paterson, M.：图上平均收益博弈的复杂度。Theor. Comput. Sci.**158**，343–359 (1996)。https://doi.org/10.1016/0304-3975(95)00188-3