arXiv:2304.05064v1 [cs.FL] 11 Apr 2023

关于正则性质的推理：一项比较研究

托马什´菲耶多尔，卢卡什´霍利克´，马丁·赫鲁什卡，阿达姆·罗加莱维奇，尤拉伊·西奇´，帕沃尔·瓦尔戈夫奇克´

布尔诺理工大学{ifiedortom,holik,ihruska,rogalew,sicjuraj,ivargovcik}@fit.vutbr.cz

摘要。 近期提出了多种新算法，用于判定正则语言的布尔组合以及交替自动机（AFA）语言的空性。这些算法尤其应用于正则表达式分析和字符串约束求解。新算法展现出显著潜力，但此前从未进行过系统比较，无论是彼此之间，还是与现有的基于（非）确定性自动机的方法的先进实现之间。本文首次开展了此类比较，并概述现有算法及其实现。我们收集了多种基准测试，其中大部分源自或涉及字符串约束求解、LTL 性质分析和正则模型检测中的实际问题，并据此评估所收集的实现。结果揭示了表现最佳的工具，并为哪些算法和实现技术最优提供了线索。总体而言，尽管某些高级算法（如反链算法和归约到 IC3/PDR）速度很快，但它们并不像有时所宣称的那样占据压倒性优势，也没有明确的赢家。根据问题来源和实现风格，最简单的基于非确定有限自动机（NFA）的技术实际上可能是最佳选择。我们的发现对于这些技术的发展以及字符串约束求解等相关领域都具有重要参考价值。

1 引言

长期以来，如何高效表示有限词的正则性质一直是研究课题，其应用和成果遍及形式化推理领域的许多方面，包括正则表达式匹配、验证、测试、建模，以及逻辑的一般判定过程。使用布尔运算及类似操作组合或生成正则性质时，会出现一些有趣的判定问题，它们是 PSPACE 完全的。这些问题包括最基本的语言空性问题（下文简称空性）。采用确定性自动机的教科书式方法在实践中饱受状态空间爆炸之苦。确定化和补运算通过指数级的子集构造完成，而合取运算的开销则是二次的。这促使研究者探索适用于非确定性自动机和交替自动机的高效算法。利用非确定性和交替性，可以分别将自动机的规模压缩一个或两个指数级。交替性在析取式的非确定性分支基础上增加合取分支，从而完全避免自动机规模膨胀。然而，从最坏情况复杂度的角度看，所获得的简洁性要以语言空性问题为 PSPACE 完全为代价。尽管如此，表示越简洁，就越有利于采用巧妙的启发式方法，

以应对最坏情况复杂度，并在实际情形中发挥作用，其核心是避免重新构造整个（非）确定性表示。

近年来提出了几种非常有前景的技术及其实现。测试 AFA 空性的最新进展出现在分析正则表达式组合以及字符串求解的背景下。其中一组技术基于将 AFA 空性归约为布尔转换系统中的可达性问题，并使用现有的模型检验算法实现（尤其是IC3/PDR [43,15], , 例如ABC [17], 、nuXMV [20], 或 IC3Ref [16], ）来解决该问题 [25,24,44,72]。来自 [66] 的最新贡献通过符号导数扩展了SMT 求解器Z3；符号导数是 Antimirov正则表达式导数的推广。Z3使用它们将正则表达式组合转换为交替/布尔自动机，并通过经典的去交替化和对接受配置的搜索，在线测试其语言空性。

稍早一些的 AFA 等价性测试算法（可转换为非空性测试）基于计算模同余的双模拟 [27]。它推广了最初的 NFA 等价性测试 [11]。同余闭包算法之前，已有通过包含关系剪枝来优化子集构造的反链算法 [74,38], , 以及首次尝试使用模型检查算法的工作，即 Impact 算法，用于处理正则性质组合的非空性问题 [57], 。最后，字符串约束求解领域催生了大量不同的字符串约束求解器。它们通过一系列巧妙的技术处理正则性质的组合，例如基于自动机、转换为其他类型的约束、对字符串长度进行推理、Parikh 映像等（例如 Z3 [59,66], CVC4/5 [7,60],Z3Str4 [9], OSTRICH [22,23], Trau [4,3] ，仅举数例）。

这些研究展示了很大的潜力，但它们是在特定且往往狭窄的背景下提出的，并且对当前最先进水平的看法各不相同。因此，它们从未得到充分的相互比较。即便是与基于（非）确定性自动机的较标准技术中最高效的实现进行比较，也很少见。字符串求解器只与字符串求解器比较，高级 AFA 空性测试只与基本的去交替化方法比较。较有意思的比较仅有： [11] 中基于 NFA-反链的方法与基于上同余的语言包含性和等价性测试之间的比较， [35], 中也是如此；以及基于基本反链的 AFA 空性方法与使用抽象解释的一个版本之间的比较 [38]。还有一些研究将自动机或字符串求解器的实现作为基准，尽管这些实现本身都是值得认可的工具，但目前在求解正则性质组合方面，无论属于哪一类，它们都不是最快的求解器。此外，所有关于求解正则性质组合的上述研究，都只采用范围狭窄且往往彼此互斥的基准测试。

要回答“该使用什么？”和“该与什么进行比较？”这些问题，并推动关于正则性质和自动机的推理领域整体发展，显然需要在有意义的基准测试上系统地比较工具和算法。因此，我们对主要算法的实现进行了比较。我们使用一个大型问题基准测试来比较这些工具，其中的问题收集自其他研究、字符串约束求解问题、正则表达式分析、正则模型检测以及系统 LTL 性质分析。我们认为，这是目前现有的最全面的基准测试。我们的重点是围绕.....的示例。

字符串求解和正则表达式分析，这也是近期大部分进展所在的领域。这些基准测试大多允许对自动机转移函数进行相对简单的表示。尽管这类示例中的字母表很大（例如，Unicode 最多包含 2^{32} 个符号），但在大多数情况下，可以通过使用字母表极小项（不可区分符号的类别）而非单个符号，将字母表大小缩减到少数几个符号。这样一来，如何有效地表示大字母表上的转移关系这一问题，就不会成为评估的主导因素；不过，在其他应用领域，例如判定 WS1S 或算术公式时，这一点至关重要。

我们获得的结果描绘了现有技术与工具的基本概貌。这些结果指出了哪些工具和方法可能表现出色，并应作为比较的基准。我们还提供了一个规模较大、类型相对多样的基准测试集，供比较之用。总体而言，结果证实，与确定有限自动机技术相比，新算法的效率有了巨大飞跃；此外，将问题归约为交替自动机的语言空性问题，也成为一种颇具吸引力的选择。另一方面，这些结果也对一些广为流传的观点以及其他研究中隐含的结论提出了质疑。例如，归约到 IC3/PDR 虽然能得到速度最快的算法之一，但并不像有时所说的那样具有压倒性优势。一些具有实际意义的基准测试类别，最适合用反链算法结合 SAT 求解器来解决。另一些类别——事实上数量出人意料地多——则适合采用面向效率、针对非确定性自动机基础算法的简单实现。我们的结果还强调，不存在普适的万灵妙方。问题的具体类型在很大程度上由其来源决定，是选择和调优求解器时必须考虑的关键因素。

2 预备知识

定义在 Σ 上的（非确定）有限自动机（非确定有限自动机）A 是一个元组 $\mathrm{A} = (Q, \Delta, I, F)$，其中$Q$ 是一个有限状态集，Δ 是一个形如 $q \xrightarrow{a} r$ 的转移集，其中 $q, r \in Q$ 且$a \in \Sigma$, $I \subseteq Q$ 是初始状态集，$F \subseteq Q$ 是终态集。A 在一个词 $w \in \Sigma^*$ 上的运行是一个序列 $p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} p_n$，其中对于所有 $1 \le i \le n$，均有 $a_i \in \Sigma \cup \{\epsilon\}$，$w = a_1 \cdot a_2 \cdots a_n$，且满足 $p_{i-1} \xrightarrow{a_i} p_i \in \Delta$或 $p_{i-1} = p_i, a_i = \epsilon$。若 $p_0 \in I$ 且 $p_n \in F$，则该运行是接受运行；A 的语言 $L(\mathrm{A})$ 是所有使 A 存在接受运行的词组成的集合。

若对于每个状态 q 和符号 a，δ 至多有一条转移 $q \xrightarrow{a} r$，则该自动机是确定的（确定有限自动机）。任何非确定有限自动机都可以通过子集构造法确定化，该方法生成确定有限自动机 $A' = (2^Q, \Delta', \{I\}, \{S \mid S \cap F \neq \emptyset\})$，其中 $S \xrightarrow{a} S' \in \Delta'$ 当且仅当$S' = \{s' \mid s \in S \wedge s \xrightarrow{a} s' \in \delta\}$。实现语言布尔运算的基本自动机构造包括交集、$\mathrm{A} \cap \mathrm{A}' = (Q \times Q', \Delta^{\times}, I \times I', F \times F')$，其中 $(q, q') \xrightarrow{a} (r, r') \in \Delta^{\times}$当且仅当 $q \xrightarrow{a} r \in \Delta$且 $q' \xrightarrow{a} r' \in \Delta'$；非确定性并集 $\mathrm{A} \cup \mathrm{A}' = (Q \cup Q', \Delta \cup \Delta', I \cup I', F \cup F')$；通过积构造实现的确定性并集，其构造与 $\cap$ 相同，只是终态为 $F \times Q \cup Q \times F$；以及补运算，它由确定化和对终态取补组成。

交替自动机。一个交替有限自动机（*AFA*）在最一般的形式下可以是一个元组 $\mathrm{M} = (\Sigma, \mathbb{P}, Q, \delta, I, F)$，其中，当用 $\mathbb{B}(X)$ 表示关于变量 X 的布尔谓词公式时：1) Σ 是一个有限字母表；2) $\mathbb{P}$ 是一个带有自由变量 α 的一元符号谓词 集合；3) Q 是一个有限状态集合；4) $\delta\colon Q \to \mathbb{B}(Q \cup \mathbb{P})$ 是

一个转移函数其中 Q 的状态仅有正向出现5) $I \in \mathbb{B}(Q)$是一个正的初始条件；以及 6) $F \in \mathbb{B}(Q)$ 是一个负的 终止/接受条件。

它可以解释为正向非确定有限自动机 $A^{\mathsf{f}} = (\Sigma, \mathrm{P}(Q), \Delta^{\mathsf{f}}, I', F')$，其状态 $c \subseteq Q$称为 A 的配置。假设对变量 Q 上的公式采用多排序解释，其中变量类型为布尔（取值为 0 和 1），而变量 α 的类型为 Σ。状态集合 $c \subseteq Q$ 被理解为赋值 $Q \to \{0, 1\}$，其中 $c(q) = 1$ 对应于 $q \in c$。一对 (c, a)， $a \in \Sigma$ 被理解为在此赋值基础上扩展了 $\alpha \mapsto a$ 的赋值。公式与配置 c 或一对 (c, a) 之间的满足关系 $\models$ 按照通常方式定义。于是，当且仅当 $(c', a) \models \bigwedge_{q \in c} \Delta(q)$，且 I' 和 F' 分别是满足 I 和 F 的配置集合时，转移关系 Δ^{f} 中包含转移 $c \xrightarrow{a} c'$。通常将 Δ^{f} 定义为仅包含最小转移；也就是说，给定 c 和 a，只有目标 c' 关于 $\subseteq$ 极小的转移 $c \xrightarrow{a} c'$ 才属于 Δ。[1] A 的语言 $L(A)$，就是 A^{f} 的语言。

交替有限自动机也可等价地解释为反向非确定有限自动机，即自动机 $A^{\mathsf{b}} = (\Sigma, \mathrm{P}(Q), \Delta^{\mathsf{b}}, I', F')$，其中 $c \xrightarrow{a} c' \in \Delta^{\mathsf{b}}$ 若 $(c, a) \models \Delta(q)$ 对每个 $q \in c$ 均成立。在给定 c' 和 a 的情况下，只需取具有 $\subseteq$-最大源 c[2]的转移（这使转移关系具有反向确定性）。

布尔自动机。 交替自动机可以通过允许在初始、终结和转移公式中使用任意布尔组合，扩展为布尔有限自动机（BFA）（初始公式和转移公式中的状态可以以负形式出现，终结公式中的状态可以以正形式出现）。注意，将AFA 扩展为 BFA 并不复杂，因为通过以下步骤，可以很容易地将一个BFA 编码为大小仅为其两倍的AFA：1)对每个 $q \in Q$，添加状态 $\bar{q}$，其具有 $\Delta(\bar{q}) = \neg\Delta(q)$，2)将 I、F、Δ中的所有公式转换为否定范式，3) 在 Δ 和 I 中将所有文字 $\neg q$ 替换为 $\bar{q}$，并在 F 中将文字 q 替换为 $\neg\bar{q}$。

AFA 转移关系的受限形式。 如上文所定义，AFA 的一般形式最为简洁，并为大多数优化留出了空间，例如 [69]。这种形式的自动机由用于 [27,69] 的 LTL 转换 [31] 生成。另一方面，只有少数算法和工具支持这种最宽松形式的 AFA。一种常见的限制（例如在 [27] 中使用）是在转移公式中将符号与状态分离，即令 $\Delta(q)$ 具有 $\varphi \wedge \psi$ 的形式，其中 $\varphi \in \mathbb{B}(\mathbb{P})$，$\psi \in \mathbb{B}(Q)$。我们将此类 AFA 称为分离式AFA。此时，转移关系可以看作一个函数$Q \to \mathbb{B}(\mathbb{P}) \times \mathbb{B}(Q)$。分离式 AFA 通常采用*DNF* 形式的状态公式 ψ，即将其写成析取范式（例如在 [33,38] 中）；此时 Δ 可以看作形如 $q \xrightarrow{\varphi} c$ 的转移集合，其中 $\bigwedge c$ 是 ψ 的一个（正）子句。

判定问题。 我们将重点讨论两个判定问题：
(1) *AFA* 空性询问给定交替有限自动机的语言是否为空。
(2) 正则性质的布尔组合的空性 (*BRE*)，询问给定为自动机或正则表达式的正则语言的布尔组合是否为空（语言可以使用 $\cap$ $\cup$和关于 Σ^* 的补集进行组合，这也涵盖了包含关系和等价性的测试[3]）。

[1] 配置中的状态被理解为一种约束。约束越少，配置所能接受的内容就越多。向约束更多的配置进行转移是无用的。[2] 反向来看，规模更大的配置具有更强的宽松性。从同一目标出发、使用规模更小的配置进行的转移是无用的。[3] $L' \subseteq L$ 是$L' \cap L$ 的空性，而等价性是$(L' \cap L) \cup (L' \cap L)$的空性。

3 现有算法与工具

在本节中，我们将概述实现交替有限自动机和 BRE 空性的现有方法和工具。

3.1 自动机转移关系的表示

最简单的形式是，自动机转移上的谓词表示字母表中的一个字母。这称为显式转移。显式形式自动机结构简单，支持底层优化，而且其复杂算法的实现也易于管理（例如计算仿真的高级算法 [62,47,21]）。先验最小项化技术以最小项构成的字母表（即无法区分的符号类）替代原字母表，使显式形式自动机在字母表较大时也能使用。然而，当最小项的数量趋于爆炸式增长时，显式形式自动机便无法扩展。

自动机的各种实现一直采用转移谓词，并以 BDD、布尔公式、位向量 SMT 理论上的公式、数字区间等形式实现。这一做法已在符号自动机 [71,28,30], 相关研究中得到系统化，其中符号谓词可以取自任意有效的布尔代数（且自动机采用分离形式）。比符号自动机更紧凑的，是 WS1S 求解器 MONA 或某些交替有限自动机（AFA）实现中使用的转移关系表示；从某种意义上说，这些表示放宽了对分离形式的限制。下面将讨论具体实现。

3.2 确定性与非确定性有限自动机

求解 BRE 的基准方法是使用确定有限自动机或非确定有限自动机。布尔运算通过第 2 节所列的经典构造来实现。自动机可以保持确定性；也可以尽可能保持非确定性，仅在进行补运算之前才将其确定化。实现高效处理的一个重要环节，通常是每隔几次运算至少对自动机进行一次最小化（例如，在正则模型检测 [12] 或某些字符串求解方法 [10,22,3]等应用中，这一点就很重要）。确定性方法通过 Hopcroft、Moore、Brzozowski 或 Huffman 算法构造最小确定有限自动机 [49,58,18,50], ; 非确定性方法则可以使用基于仿真 [62,21,42,51,47] 或双仿真 [68,61,45] 的归约方法。仿真归约效果显著得多，但代价也高得多。许多库都实现了确定有限自动机和非确定有限自动机。这里我们选取一些有代表性的例子。

首先，ENFA 是最简单的工具，是我们自行实现的 NFA。它使用带有最小项化的显式形式自动机。该实现使用 C++ 编写，注重效率，但没有进行大量优化（大致而言，从一个状态出发的转移存储在一个两层数据结构中：第一层按符号划分并排序，第二层按目标状态排序）。它使用了 VATA 树自动机库 [53] 中的现成实现，该实现采用最新一代的仿真计算算法 [62,21,47] 之一（通过使用分区关系数据结构实现了良好的效率），具体实现的是 [47]。[4]

以 Java 编写的金砖国家自动机库 [6],常被视为比较基准 [6]。它主要使用确定性自动机和符号表示

[4] 在我们的实验中，仿真仅在解析之后使用，对整体影响极小。

来表示基于字符范围的转移关系。它以 Java 编写，且经过了相对优化。

以 C# 编写的 AUTOMATA 自动机库 [70], 实现了由有效布尔代数参数化的符号非确定有限自动机/确定有限自动机。我们使用其默认的 BDD 代数。AUTOMATA 已经开发多年，积累了许多用于处理符号自动机的优化和新颖技术（例如，优化的最小化算法 [29]）。

以 C 编写的 MONA [41], 是最具影响力且经过高度优化的确定性自动机实现。它专门用于判定 WS1S 公式，其中除了布尔组合外还包括量化。该判定过程会生成确定有限自动机，其在大型位向量字母表上具有复杂的转移关系。为此，MONA 使用紧凑的转移关系表示：用一个 MTBDD 表示从某个状态出发的所有转移，并将目标状态存储在其叶节点中。MONA 只能表示确定有限自动机，因此始终会隐式地进行确定化。

以 C 编写的 VATA [53], ++是一个实现非确定树自动机的库。由于非确定有限自动机是树自动机的特例，我们可以将其用于实现显式形式非确定有限自动机的基本构造。它经过了相对充分的优化。我们将其纳入比较，是因为它能快速实现反链包含性检查 [12,46],；对于非确定有限自动机，这归结为对 [33] 的包含性检查。

3.3 交替自动机

去交替化。AFA 空性的基本方法是去交替化，即转换为 NFA，可以采用正向方法 A^{f} 或反向方法 A^{b}，然后测试所得NFA的空性。这两个NFA 都是通过NFA子集构造的一种变体构造的。我们不知道有任何工具使用纯粹的去交替化，并且我们认为它不会具有竞争力。然而，正向算法是Z3 中所使用方法的基础 [65]，该算法通过一种新颖的符号导数构造在线运行（字符串约束求解器段落中也讨论了这一点）。

基于插值的抽象精化。在字符串求解和正则表达式处理的背景下，人们开始尝试将模型检测算法用于 AFA 空性。据我们所知，最早的尝试出现在 [37], ，其中使用[56]的基于插值的算法求解正则约束的合取。 [52]也采用了基于插值的抽象精化算法 Impact，该算法由[57], 提出。这项工作聚焦于一个更一般的问题：求解在无限数据域上的数据词的 AFA 空性（该数据域可以关联数据变量的过去值和当前值）。他们的工具 JALTIMPACT [2] （使用 Java 编写）也被纳入我们的比较，并且可以在我们的基准测试上运行。

归约为可达性和*IC3/PDR*。 [72] 的工作首次将字符串约束（主要是 BRE）转译为布尔转换系统（电路）中的可达性问题，随后使用模型检查器nuXmv [20]进行求解。这实际上是首次将 AFA 空性归约为布尔转换系统（BTS）中的可达性问题。

下面简要概述该归约的基本原理。对于 AFA A，正向BTS 具有配置，这些配置是对 Q的布尔赋值；初始配置和最终配置分别满足 I 和 F，而转移由公式 $\Phi^{\mathsf{f}}_{\Delta} :\bigwedge_{q\in Q} q \to [\Delta(q)]'$给出。在此，我们使用 $[\varphi]'$ 表示通过将每个状态 q 替换为其加撇版本 q'而从 φ 得到的公式，并且我们还将 $[c]'$ 表示为

配置 c 的撇号版本 $\{q' \mid q \in c\}$。一个配置 c 的后继 是满足以下条件的任意配置 $\bar{c}$： $[\bar{c}]'$ 满足 $\exists Q \exists \alpha \Phi^{\mathsf{f}}_{\Delta} \wedge \bigwedge_{q \in C} q$（符号变量alpha 属于位向量排序）。可达性随后定义为后继关系的传递自反闭包，而可达性问题询问终结配置是否可从初始配置到达。当且仅当 A 非空时，情况成立。 [72] 使用了正向归约。或者，对于 A，后向 *BTS* 具有满足 F的初始配置、满足 I 的终结配置，以及由公式 $\Phi^{\mathsf{b}}_{\Delta}$ 给出的后继关系：$\bigwedge_{q \in Q} q' \rightarrow \Delta(q)$。

[25,24] 的工作采用 IC3/PDR [43,15],，并在 IC3Ref [16], 中实现了结合后向 BTS 归约的方法，以求解 BRE 的空性问题，取得了非常令人鼓舞的结果。然而，他们的实现是专有的，未向公众开放。 [44], 采用了类似的方法：将字符串约束转换为多带 AFA，再通过前向转换将其转换为 BTS，并交由 IC3/PDR 求解，使用的工具包括 nuXmv [20] 和 ABC [17]。 [69] 的结果似乎表明，后向转换效果更好； [25,24] 中的比较也得出了相同结论。在该比较中，基于前向归约的字符串求解器 Sloth [44], 比基于后向归约的 Qzy 慢得多。在此比较中，我们还纳入了自己基于模型检查器 ABC 实现的后向归约 C++工具 BWIC3。

反链。 [74]中介绍的反链算法是解决 BRE的首次突破。它们利用由子集构造法的变体构造出的自动机的状态之间的包含关系来剪枝构造过程。它们被用于检验非确定有限自动机的语言全性与包含关系，以及 AFA 空性。AFA 空性具体基于即时搜索 A^{f} 的一个接受状态或 A^{b} 的一个初始状态。包含关系会剪去已发现的、比其他状态更大的状态（对于反向算法则剪去更小的状态）。

反链算法在许多工作中得到了增强和推广，例如通过基于仿真的包含关系 [5,33], 或 [38]中的反例引导的抽象精化，实现了更激进的剪枝。在本比较中，我们纳入了VATA树自动机库中实现的非确定有限自动机包含性检查 [53]。我们还尝试了一种由学生实现的反链 AFA 空性检查 [38] ，该实现使用抽象精化（原始实现已不再维护，我们也无法运行它）。然而，由于无法达到具有竞争力的性能，我们将其排除在比较之外。性能不佳的一个原因可能是，最简单形式的 AFA——显式形式的 DNF（原始版本 [38]所使用的形式）——在我们的示例中构造起来可能效率过低且代价高昂，部分原因在于 AFA 空性基准测试所诱导的大量最小项。

我们用 C++ 实现了 [33] 的反链式 AFA 空性测试，该测试与 SAT 求解器紧密集成，以处理字母表较大的通用形式 AFA。我们将其称为 ANTISAT。下面简要说明其原理。它本质上实现了前一段讨论的反向 BTS 可达性测试。配置 c 由合取式 $\phi_c = \bigwedge_{q \in Q \setminus c} \neg q$表示。注意， ϕ_c 由 c的向下闭包满足，其中包含所有被 c包含（或蕴含）的配置。为了计算由 ϕ_c表示的配置的前驱，会调用 SAT 求解器（即 MiniSAT [34]）来处理公式 Φ： $\Psi^{\mathsf{b}}_{\Delta} \wedge \phi_c \wedge \psi_{\mathsf{Ach}}$。这里， ψ_{Ach}会从解中排除所有已发现的配置。它是由子句 ϕ_c 构成的合取式：对于每个先前发现的配置 $\overline{c}$，均有 $\bigvee_{q \in Q \setminus c} q$ 。SAT 求解器

会发现一个满足赋值 e，并将其转换为一个新配置 $c' = Q \cap e$（也就是说，构成位向量 α 的符号位的取值会从e中省略）。除非 c' 是初始配置，否则会将其加入队列以便进一步计算前驱，并立即通过增量 SAT 求解接口将其作为子句$\phi_{c'}$加入 ϕ_{Ach}。最后，我们只关心 c 的极大前驱，因为非极大前驱会被它们包含。我们通过直接操作 SAT 求解器的内部结构来确保 c 的极大性：在决策点，强制 SAT 求解器优先选择将状态变量赋值为 1 的决策。

基于同余的双模拟向上技术。后来出现的一类算法（此处称为向上算法），通过检验交替有限自动机配置之间的双模拟来判断等价性，并利用基于同余的向上技术剪枝搜索空间。最初用于判定非确定有限自动机等价性的算法 [11] 被扩展用于交替自动机空性检验 [27]。这些算法与反链算法很接近。 [11] 表明，从理论上说，向上技术的剪枝潜力与反链算法相同或更大。基于同余的向上技术的一个缺点是，需要耗费大量计算来求值同余闭包。 [35]开展的更广泛实验表明，反链算法速度更快，但随机生成的字母表较小且转移关系非常稠密的自动机除外。我们将 [27] 的 AFA 空性检验 Java 实现也纳入比较（空性检验可化简为与一个平凡的空 AFA 进行等价性检验），并将其称为 BISIM。据我们所知，[36] 和 [11], 提供的其他向上算法实现都是专用程序，只能判定两个 NFA 是否等价，因此我们只能在基准测试中极少数实例上运行这些程序。

3.4 字符串约束求解器

有数十种字符串约束求解器在不同程度上实现了对正则性质组合进行判定的支持。字符串语言十分丰富，而 BRE并非这些求解器的绝对优先事项，因此它们在处理 BRE时通常不如专用工具。但字符串求解器实现了大范围的独特技术和实用启发式方法，这些方法可能在特定实例中发挥作用。具有最成熟实现的求解器代表（也被文献中的大多数比较研究所采用）是 Z3 [59,65]和CVC5 [60,7]。CVC5 主要通过重写规则求解 BRE。最近， [65] 将一种基于Antimirov 导数自动机构造的方法扩展到了符号自动机和扩展正则表达式。实质上，该构造生成一个符号AFA/BFA，并在执行前向去交替化的同时动态检查其是否为空。 [65] 表明，在求解 BRE 方面，它显著高效于其他 SMT 求解器（包括 CVC5）。

3.5 Other Approaches and Tools

尽管我们认为自己收集了现有算法和工具中具有代表性的一部分，但并未收集所有有趣的实例。有些实例无法获取，有些难以运行或难以准备输入，还有些似乎已在其他研究的实验中得到覆盖。将这些工具和算法纳入比较仍可能颇有意义，我们将其留待未来工作（我们计划继续扩充工具库和基准测试集）。具体而言，在 [25], 的比较中使用的工具 DPRLE [48], 似乎大多不如 IC3/PDR

在 Qzy 中实现的方法，不过并非始终如此。 [35] 中使用的非确定有限自动机反链技术和 up-to 同余技术实现得似乎很高效，其非确定有限自动机反链包含性检查速度是 VATA 的两倍。 [11] 的 up-to 同余非确定有限自动机等价性检查也可能很快（[11] 和 [35] 报告的结果存在一定冲突）。有许多非确定有限自动机/确定有限自动机库，例如 BRICS 的 C 语言替代实现 [55]，或 [26] 的符号非确定有限自动机 Java 实现。ALASKA [32] 可能包含有趣的反链算法实现，但该项目已不再维护，也无法获取。我们的比较缺少一种用于合取范式形式显式交替有限自动机、基于反链的基本去交替化实现；如果没有被大量最小项拖累，它或许能凭借简单快速的数据结构取得良好性能，类似于我们的 ENFA。

4 基准测试

我们收集了尽可能全面的基准测试，既整理了以往工作中使用的示例，也生成了一些我们自己的示例。相关内容可在 [1]获取。我们主要关注近年来AFA 和 BRE空性求解方面发展最为集中的领域，即字符串约束求解，以及正则表达式的分析和用于文本分析与过滤的正则表达式。这里的原子正则性质主要以基于Unicode字符类的正则表达式形式给出。字母表很大，但最小项的数量通常较少或适中。对于我们来自正则模型检测的示例，这一点同样成立。因此，对大字母表上的复杂转换关系进行符号化处理并非绝对关键，实验可以专注于空性检查的主要算法。出于这一原因，我们不包含求解 WS1S [19]，（Mona 的主要目标）或带自动机的 Presburger 算术 [13,73], 的基准测试，因为在这些问题中，处理符号字母表的技术不可或缺。专门针对这类问题的技术值得进行单独研究。我们的基准测试中，符号字母表表示仍然相当重要的部分包括：来自（若干）LTL 性质组合、字母表由原子命题集合构成的AFA，以及将字符串约束问题转换为具有复杂多轨字母表的 AFA。[5]

正则表达式的布尔组合。 这组BRE 包含基准测试，我们可以在其上运行所有工具，包括那些基于非确定有限自动机和确定有限自动机的工具。它们的最小项数量较少到中等（平均值约为 30 个，最多超过一百个）。

b-smt 包含来自 Norn和 SyGuS-qgen 的 330个字符串约束，这些约束收集于 SMT-LIB 基准测试 [8], , 属于 BRE。这些约束也用于比较 [66] 中的 SMT- 求解器。

[5] 我们没有尝试生成纯粹随机的问题。首先，纯粹随机生成的自动机（例如由 [67] 生成的自动机）似乎具有不同于来自实际问题的自动机的特征（例如见 [12,35]）。其次，尽管可以通过由三个简单参数控制的生成器来生成随机非确定有限自动机，这些参数提供了一个可管理的参数值空间，覆盖所有非确定有限自动机，但目前尚不清楚如何以类似方式生成随机交替有限自动机或BRE。另一方面，我们确实纳入了一个基于随机生成的LTL公式的基准测试，我们认为该基准测试相对接近现实的LTL规范。

b-hand-made 包含来自 [66] 的 56 道难度较高的手写题，涉及扩展了交集和补运算的正则表达式成员判定问题。这些问题涵盖 (1) 日期和密码问题，(2) 布尔运算与连接和迭代相互作用的问题，以及 (3) 具有指数级确定化复杂度的问题。

b-armc-incl 包含来自 [12] 的抽象正则模型检验工具运行结果的 171 个语言包含问题（验证 bakery 算法、冒泡排序和生产者—消费者系统）。这些示例也用于 [35,11]。

b-regex 包含 500 个问题，其获取方式与 [27,69], 类似，形式为 $r_1 \wedge r_2 \wedge r_3 \wedge r_4 = r_1 \wedge r_2 \wedge r_3 \wedge r_4 \wedge r_5$，其中每个 r_i 都是 75 个正则表达式中的一个[6]，这些正则表达式来自 RegExLib [63]，并经过选择以确保 $r_1 \wedge r_2 \wedge r_3 \wedge r_4 \wedge r_5$ 非空。该基准测试受到垃圾邮件过滤的启发：我们希望测试新过滤器是否 r_5 会向现有过滤器中添加任何内容。我们将该问题转换为包含问题 $r_5 \subseteq r_1 \wedge r_2 \wedge r_3 \wedge r_4$，并保留原始形式供 Bisim 使用，因为它要求判断等价性。b-

p_{aram} 有 8 p参数化 p个问题。其中四个来自 [37]: (1) [a-c]a[a-c]{$n+1$} ∩ [a-c]a[a-c]{n}（长字符串），(2) $\bigcap_{i=1}^{n}$([0-1]{$i-1$}0[0-1]{$n-1$}0[0-1]{$n-i$}α_i)|([0-1]{$i-1$}1[0-1]{$n-1$}1[0-1]{$n-i$}α_i)（指数级分支），(3) $\bigcap_{i=1}^{n}$.*(.{p_{10+i}})+α_i（指数级路径 1），以及 (4) $\bigcap_{i=1}^{n}$.+α_i0(.{p_{10+i}})+（指数级路径 2），其中 α_1、. . . 、α_n是互不相交的字符类，p_j 是第 j 个素数。另有四个来自 [25]: (5) ^.[01]*.1.[01]{n}.$ \ ^.[01]*.0.[01]{n-1}.$（可满足的差集），(6) ^.[01]*.1.1.[01]{n}.$ \ ^.[01]*.0.[01]{n+1}.$（不可满足的差集），(7) ^.[01]*.1.[01]{n}.$∩^.[01]*.0.[01]{n-1}.$（可满足的交集），以及 (8) ^.[01]*.1.[01]{n}.$∩^.[01]*.0.[01]{n}.$（不可满足的交集）。对于 (1)，我们选择了 $n \in \{50, 100, \ldots, 500\}$；对于 (2)-(4)，选择了 $n \in \{2, 3, \ldots, 60\}$；对于 (5)-(8)，选择了 $n \in \{50, 100, \ldots, 100\}$。

4.1 AFA 基准测试

第二组示例包含难以转换为 BRE 的 AFA。在这里，我们只能运行处理一般 AFA 空性的工具。这些基准测试中的一些还具有大量的最小项（很容易达到数千个），且 AFA 转移函数中的公式很复杂，因此将它们转换为分离析取范式或显式形式等受限形式可能代价高昂。这似乎也是我们的实现 [38] 无法与之竞争的主要原因。

a-ltlf-patterns 源自将有限轨迹上的线性时序逻辑公式（LTL_f）转换为 AFA [31]。这 1699 个公式来自 [54][7]，它们表示常见的 LTL_f模式，可分为两组：（1）7 个参数化模式（每组 100 个）；（2）由更简单的 LTL_f模式随机生成的合取式（999 个公式）。

a-ltl-rand 包含由 [69] 的随机生成器生成的 300 LTL_f 个公式。该生成器遍历 LTL 文法的语法树，并由以下内容控制。

[6] https://github.com/lorisdanto/symbolicautomata/blob/master/benchmarks/src/main/java/regexconverter/pattern%4075.txt

[7] https://drive.google.com/file/d/1eOYGvm3C8sQ-9iyfZ8qx42K54hgrFNTC

变量数量、联结词概率、最大深度和平均值深度。我们以经验方式设定了参数，以尽可能生成对比求解器难以处理的示例（公式包含 6 个原子命题，最大深度为 16）。

a-ltl-param 包含一对手工编写的参数化 LTL_f公式（每个包含 160 个公式），用于 [27,69]: *Lift* [40] 描述了一个在参数化数量楼层上运行的简单升降机，而*Counter* [64] 描述了一个按参数取模递增的计数器。a-ltlf-spec [54] 包含 62 LTL_f个用于描述现实系统的公式，由波音 [14] 和美国国家航空航天局 [39]使用。这些公式表示用于设计波音 AIR 6110 车轮制动系统以及设计美国国家航空航天局 NextGen 空中交通管制（ATC）系统的规范。

a-sloth 4062个交替有限自动机空性问题，字符串求解器 Sloth将字符串约束 [44] 归约为这些问题。交替有限自动机具有编码布尔运算和转导的复杂多轨转移，以及一种特殊的轨迹同步机制，需- 要复杂的初始和最终条件。

a-noodler 13840 个 AFA空性问题，这些问题对应于字符串求解器 Noodler 在 [10] 中解决的某些子问题。AFA 的创建方式与 a-sloth 的 AFA类似，但它们针对不同的输入自动机，对一组不同的特定操作进行编码。

5 比较

我们在Debian GNU/Linux 11 上进行了实验，使用英特尔酷睿3.4GHz处理器、8 个 CPU核心和 20GB 内存。所有实验均在 60 秒超时限制下运行（增加超时时间并未产生显著影响）。更多详细信息以及包含完整基准测试的虚拟机可在 [1] 获取。

基准测试基础设施。 初始难点在于，各工具要求的输入格式和自动机形式各不相同，基准测试也有不同的格式。我们将所有基准测试转换为内部 AFA 格式，再由此生成 AFA 处理工具 JALTIMPACT 、 BWIC3 、 ANTISAT 和 BISIM 支持的格式，或者为工具扩展解析器。BRE 基准测试来自各种来源。我们首先将它们转换为一个主文件，该文件指定原子 NFA 的布尔组合，每个原子 NFA 都存储在单独的文件中。随后生成供 Z3 和 CVC5 使用的 SMT-LIB 格式。对于 b-hand-made、b-param 和b-smt，原子自动机通过 BRICS 的解析器从正则表达式转换而来；而对于b-regex，由于其中的正则表达式包含 BRICS 不支持的特性，我们使用 BISIM 的解析器。b-smt 和 b-hand-made 需要先从 SMT-LIB 转换为正则表达式。对于b-armc-incl，原子自动机直接以 NFA 的形式提供，并被转换为各个 BRE 求解器所需的格式（我们还为其中一些求解器编写了解析器），以及供 AFA 求解器使用的内部 AFA 格式。我们为每个 BRE 求解器都扩展了一个主文件解释器，用于读取从生成的、针对求解器的文件中提取的 NFA/DFA（SMT 求解器除外，它们读取 SMT-LIB）。需要指出的是，由于内部结构方面的一些困难，目前我们无法在b-armc-incl上运行 BRICS；此外，由于缺少能将复杂正则表达式和纯 NFA 转换为 SMT 格式的转换器，我们不会在 b-regex 和b-armc-incl上运行 Z3 和 CVC5。

tool	a-ltl-rand			a-ltl-spec			a-ltlf-patterns			a-noodler			a-sloth			a-ltl-param		
bwIC3	**0.1**	**0.1**	**0**	0.1	0.1	0	**0.1**	**0.1**	**0**	**0.1**	**0.1**	**3**	**1.3**	**0.1**	**34**	**25.4**	**0.6**	**134**
BISIM	4.4	1.0	8	32.9	60.0	32	37.0	60.0	1013	31.6	26.4	6644(8)	17.5		1.5	1087(10)	58.2	60.0	308
JaltImpact	7.9	2.3	12	2.4	1.4	0(1)	4.0	2.8	0	3.8	1.8	186	24.1	15.4		958	47.0	60.0	205
Antisat	18.3	0.1	84	**0.0**	**0.0**	**0**	31.0	60.0	868	0.4	0.0	57	14.9		0.0	991	58.3	60.0	310

tool	b-armc-incl			b-hand-made			b-regex			b-smt			b-param		
BWIC3	5.2	1.1	1	0.4	0.1	0	**0.2**	**0.1**	**0**	0.1	0.1	0	44.9	60.0	191
BISIM	28.5	9.5	72	11.2	1.0	8	3.8	1.3	15	2.5	2.5	0	55.4	60.0	240
金砖国家		-		3.9	0.4	3	5.8	0.8	40	0.3	0.3	0	52.7	60.0	228
CVC5		-		27.4	0.8	10(15)		-		0.8	0.2	1	48.6	60.0	208
自动机	3.5	0.4	9	0.2	0.2	0	0.2	0.2	0	0.2	0.2	0	46.3	60.0	161(42)
JaltImpact	30.9	24.6	63	11.1	3.6	5	12.2	2.4	48	3.5	3.5	0	57.8	60.0	252
Antisat	42.8	60.0	118	1.4	0.0	1	9.3	1.4	45	0.0	0.0	0	39.0	60.0	147
MONA	28.5	44.1	43	27.3	0.1	22(3)	41.0	60.0	15(298)	1.5	0.0	8	44.9	60.0	25(169)
ENFA	**1.9**	**0.8**	**0**	**0.1**	**0.0**	**0**	**0.2**	**0.1**	**0**	**0.0**	**0.0**	**0**	44.6	60.0	143(51)
VATA	2.6	3.4	0	**0.1**	**0.0**	**0**	2.1	0.2	10(1)	**0.0**	**0.0**	**0**	37.8	60.0	155(1)
Z3		-		3.9	0.0	2		-		0.4	0.0	2	**32.0**	**48.1**	**129**

表 1. 交替有限自动机和 BRE 基准测试汇总。表格列出了 (i) 平均值、(ii) 中位数，以及 (iii) 超时和错误的数量（括号内）。获胜者以粗体突出显示。

测量数据。 我们将分别展示 BRE（运行所有工具）和 AFA 空性（运行 BWIC3、ANTISAT、BISIM 和 JALTIMPACT）得到的结果。我们还将应用实例的结果与参数化手工构造实例的结果分开呈现。

表 1 总结了对基准测试进行评估所得的统计数据。表中列出了：(i) 平均运行时间，(ii) 中位数运行时间，以及 (iii) 超时和错误的数量（大多是工具耗尽内存、内存分配失败或遇到段错误所致）。少数错误（例如 CVC5 或 BISIM 中的错误）是由输入中包含不受支持的特性导致的。随后，我们在图 1 的仙人掌图中直观展示了各工具的性能。对于每种工具，图中展示了该工具在各项基准测试上的运行进度： y 轴表示基准测试所用时间的累计值， x 轴上的各个示例则按该工具的运行时间排序。超时的情况不予显示。附录中还给出了一组散点图，对每项基准测试中表现最好的三种工具进行比较。

最后，我们比较了各工具在参数化基准测试 a-ltl-param 和 b-param上的表现。结果见图 2。每张图都展示了特定参数取值递增时的运行时间，参数值沿 x 轴递增。

5.1 讨论

根据测量结果，我们得出以下几点观察。

首先，将全性（既可在 AFA上运行，也可用于BRE 空性）与最稳定的良好性能相结合的工具是 bwIC3。它在大多数AFA空性基准测试中占据优势，在BRE 基准测试上表现出色或非常出色，并且经常在参数化示例中脱颖而出。此外， [25] 中报告的测量结果表明，反向 BTS 归约仍具有更大的潜力。这一点尤其可以从我们在参数化基准测试diff-sat、diff-unsat、inter-sat 和inter-unsat上的结果比较中看出。我们的实现与 [25] 在diff-sat 上的结果一致，在inter-sat上部分一致；在 diff-unsat上呈现出较差的趋势，在 inter-unsat上的趋势则差得多。一个可能的原因是底层模型检查器不同：我们的实现使用 ABC [17] ，而 [25]使用的是 ICRRef [16] 。 [25] 然而并未开箱即用地使用IC3Ref，高效利用 IC3Ref 并非完全 trivial。

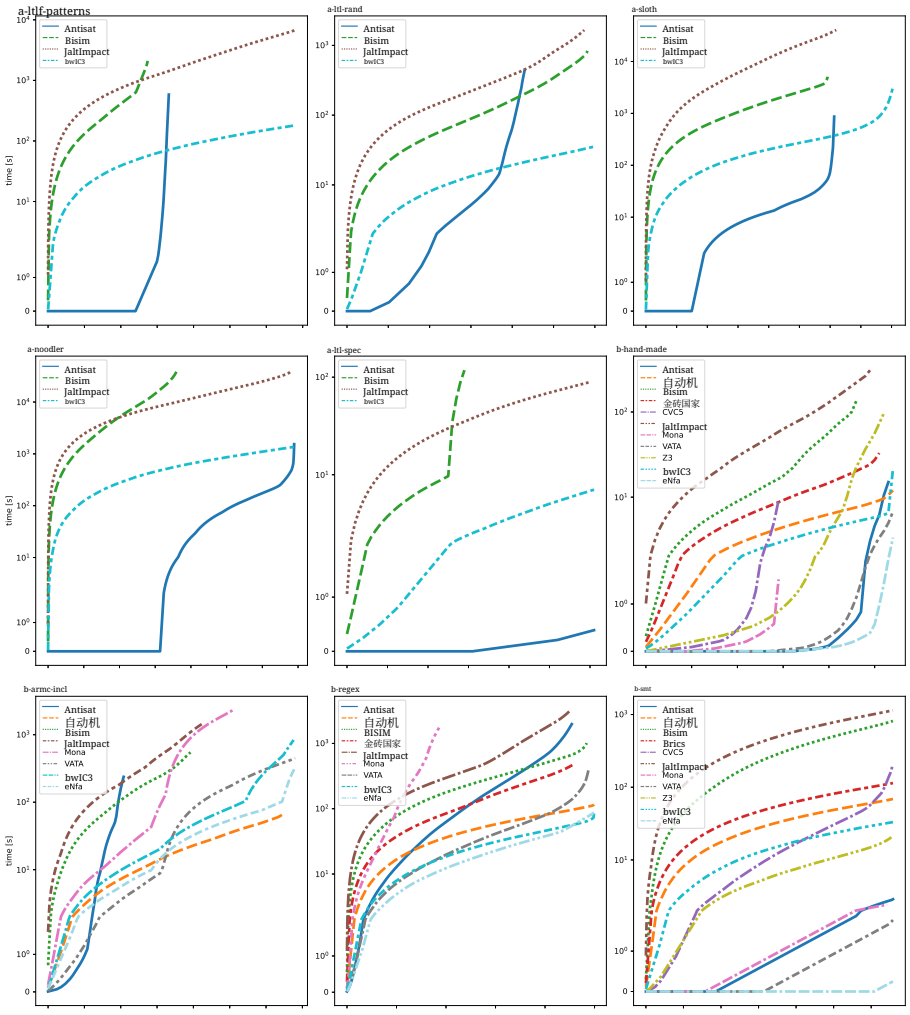


图 1. AFA 和 BRE 基准测试的仙人掌图。 y 轴表示基准测试所用的累计时间，采用对数刻度；x 轴上的基准测试按照各工具的运行时间排序。

其次，在与应用相关的 BRE 上（除 b-param 中的参数化示例外的所有 BRE），结果颇令人意外地有利于主要基于相对基础的 NFA 算法的工具。整体表现最佳的是所有工具中最简单的一个，即我们实现基本 NFA 构造的工具 ENFA。需要强调的是，尽管设计时考虑了效率，它并未经过优化，而其简洁性也为进一步优化留出了很大空间。VATA 的表现与 ENFA 接近，它在 b-armc-incl 和 b-regex 上使用反链包含性检查（ENFA 的显式补运算比 VATA 的反链方法更快，这表明这些包含性基准测试并不特别困难）。VATA 专用于更一般的树自动机，这或许使其表现.....

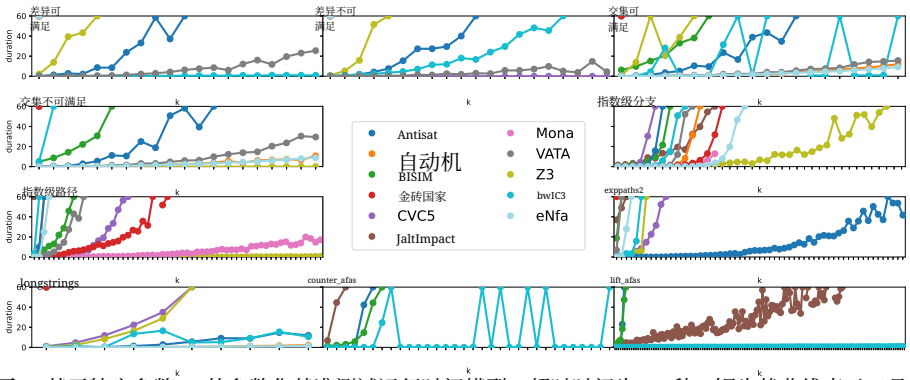


图 2. 基于特定参数 k 的参数化基准测试运行时间模型，超时时间为 60 秒。锯齿状曲线表示工具在求解 $k-1$ 和 $k+1$ 的基准测试时，因某些 k 而未能完成基准测试。为简洁起见，我们仅绘制模型，直到它们开始持续失败为止。

可能会导致不必要的开销。AUTOMATA 的表现也很好。它使用了比 ENFA 稍微更先进的算法（例如对差集进行惰性求值，不过没有使用反链剪枝）。它使用 BDD 对转移函数进行符号表示，可能在这里并没有带来多少优势。这一结果对这样一种观点提出了质疑：将复杂问题（例如字符串约束求解中产生的问题）转换为 AFA，以便使用 AFA 求解器的复杂机制，显然是一种万灵丹。将计算组织成更小的 NFA 操作，同时还可以对部分结果进行最小化并重复利用，并采用更简单、因而更灵活的 NFA 技术，或许是更好的策略（例如，这种方法在我们最近的原型字符串约束求解器 [10]中似乎运行得非常好）。

我们基于反链算法和 SAT 求解器实现的 AFA 空性测试工具 ANTISAT 表现颇为有趣。从仙人掌图可以看出，除了在a-ltlf-spec上绝对领先外，它在很大一部分其他 AFA 空性基准测试中也显著快于其他工具，但在其余测试中则表现不佳。它占优的例子通常是结构类似于带有长柄的套索（或多个套索）的自动机。另一个反链算法实现——VATA 中的 NFA/NTA 包含性检查——也表现良好。这些结果共同体现了反链算法的整体实力。

SMT 字符串约束求解器在与实际应用相关的基准测试中并非表现最佳，但具有竞争力（尤其是 Z3），并且在某些参数化案例中胜出。这可能是因为 SMT 求解器特有的各种启发式方法发挥了作用，尤其是将一种约束重写为另一种约束的重写技术。例如，Z3 似乎借助重写，将 指数级路径 求解为序列理论中的子字符串约束。总体而言，参数化示例的测量结果强调了这样一个事实：没有任何算法能够普遍优于其他算法，其相对性能可能会因输入类型的不同而发生巨大变化。

尽管其他工具表现平平，部分原因可以归结为它们关注的问题类型不同，或采用了过时的底层技术，而且各自也都有可取之处，但在比较解决此类问题的工具时，不应将它们作为基准工具。Mona 针对不同的场景进行了优化——即包含许多最小项的位向量复杂字母表——

但其性能受制于隐式确定化，而且在我们的场景中，符号表示带来的开销可能也是原因之一。它还经常因 BDD 节点超出 32 位地址空间而无法继续运行。BRICS 也有类似问题，它同样总是进行确定化。相较于 [11,27] 中报告的良好 up-to 算法结果，BISIM 的低性能令人意外。这与 [35] 的结果更为一致：在那些更实用的示例中，up-to 算法并未胜过反链。不过，我们的结果并不直接与 [27] 本身的结果矛盾，因为该研究没有与此处指出的快速工具进行比较，而且在很大程度上依赖参数化基准测试和随机基准测试。我们也不能排除输入准备方式不适合该工具的可能性。例如，BISIM 要求将输入转换为分离式交替有限自动机，而这一转换并非易事。似乎有必要对此作进一步研究，并与其他一些 up-to 技术实现进行比较。考虑到 JaltImpact 面向的是另一类系统——数据词上的交替有限自动机——它在 BRE 和 AFA 空性任务上缺乏原始速度优势，也在意料之中。图中呈现出的稳定趋势表明，针对 BRE 和 AFA 空性优化的、基于插值的抽象精化实现或许具有潜力。

主要结论。 将 AFA空性反向归约为 BTS 可达性，并结合 IC3，速度非常快且适用性极强，在几乎所有基准测试中都展现出非常出色的性能。然而，在与真实世界应用相关的 BRE 上，简单的非确定有限自动机算法实际上往往具有最佳的原始性能，其中非确定有限自动机的最简单实现表现最佳。反链算法同样表现良好，甚至在特定类型的 AFA上显著优于其他算法。这些算法似乎是应当采用的工具。对结合 IC3 的反向 BTS 归约、反链算法以及基本非确定有限自动机算法进行合理实现，也应当作为比较的基线。

基于确定有限自动机的 Mona 和 金砖国家，以及专注于数据词而非纯正则性质的 JALTIMPACT，均未达到最佳工具的性能。此外，BISIM 未能证实 up-to 算法的能力。SMT 求解器，尤其是 Z3，具有竞争力，但不能被视为最先进水平。

总体而言，基准测试的具体类型和来源是影响工具性能的决定性因素，这一点在参数化基准测试中尤为明显。

通往有效性的线程。我们的结果应谨慎看待，因为实验本身存在固有的误差空间。尽管我们尽可能力求公平，但由于我们并不熟悉每一种工具，第 5 节开头所述的格式转换以及不同类型自动机之间的转换，可能给实验带来了偏差。各工具使用不同语言编写，有些工具还带有参数，我们可能没有以最优方式使用这些参数（我们采用工具的默认设置）；对于库而言，我们也可能没有选择最优的函数组合。此外，我们没有测量内存峰值，而这在例如将工具部署到云端时尤其值得关注。不过，我们确信主要结论有充分依据，而且实验总体上提供了良好的概况。任何人都可以查验或改进整个实验 [1]。

16 位作者因篇幅过长而省略

参考文献

1. 实验复现包及补充材料，https://www.fit.vutbr.cz/research/groups/verifit/tools/afa-comparison/ 2. JaltImpact，https://github.com/cathiec/JAltImpact 3. Abdulla, P.A., Atig, M.F., Chen, Y., Diep, B.P., Hol´ık, L., Rezine, A., R¨ummer, P.: Trau：用于字符串约束的 SMT 求解器。载于：2018 年计算机辅助设计中的形式化方法会议（FMCAD 2018），美国得克萨斯州奥斯汀，2018 年 10 月 30 日至 11 月 2 日。第 1–5 页。IEEE（2018）。https:// doi.org/10.23919/FMCAD.2018.8602997，https://doi.org/10.23919/FMCAD.2018. 8602997 4. Abdulla, P.A., Atig, M.F., Diep, B.P., Hol´ık, L., Janku, P.: 无链字符串约束。载于：验证与分析自动化技术——第 17 届国际研讨会，ATVA 2019，台湾台北，2019 年 10 月 28 日至 31 日，论文集。《计算机科学讲义》，第 11781 卷，第 277–293 页。Springer（2019）。https://doi.org/10.1007/978 3 030 31784 3_16 10 1007 978 3 030 31784 3_16----，5 https://doi.org/ . / - - - - 。Abdulla, P.A., Chen, Y.F., Holık, L., Mayr, R., Vojnar, T.: 仿真与反链相遇。载于：TACAS'10。《LNCS》，第 6015 卷，第 158–174 页。Springer（2010）6. et al, A.M.: Brics 自动机库，https://www.brics.dk/automaton/ 7. Barbosa, H., Barrett, C., Brain, M., Kremer, G., Lachnitt, H., Mann, M., Mohamed, A., Mohamed, M., Niemetz, A., N¨otzli, A., Ozdemir, A., Preiner, M., Reynolds, A., Sheng, Y., Tinelli, C., Zohar, Y.: cvc5：一种多用途、具有工业级实力的 SMT 求解器。载于：系统构建与分析的工具和算法。第 415–442 页。Springer International Publishing，Cham（2022）8. Barrett, C., Fontaine, P., Tinelli, C.: 理论模可满足性库（SMT-LIB）。www.SMT-LIB.org（2016）9. Berzish, Murphy: Z3str4：字符串理论求解器。博士论文（2021），http: //hdl.handle.net/10012/17102 10. Blahoudek, F., Chen, Y.F., Chocholat´y, D., Havlena, V., Hol´ık, L., Leng´al, O., S´ı˘c, J.: 字方程与正则约束的协同作用。载于：Chechik, M., Katoen, J.P., Leucker, M.（编）形式化方法。第 403–423 页。Springer International Publishing，Cham（2023）11. Bonchi, F., Pous, D.: 通过提升至同余的双模拟检验非确定有限自动机等价性。载于：POPL'13。第 457–468 页。《ACM 计算逻辑汇刊》（2013）12. Bouajjani, A., Habermehl, P., Hol´ık, L., Touili, T., Vojnar, T.: 基于反链的非确定有限树自动机的普遍性与包含性测试。载于：自动机的实现与应用。第 57–67 页。Springer Berlin Heidelberg，柏林，海德堡（2008）13. Boudet, A., Comon, H.: 丢番图方程、Presburger 算术与有限自动机。载于：Kirchner, H.（编）代数与编程中的树——CAAP'96，第 21 届国际学术讨论会，瑞典林雪平，1996 年 4 月 22–24 日，论文集。《计算机科学讲义》，第 1059 卷，第 30–43 页。Springer（1996）。https://doi.org/10.1007/3 540 61064 2_27 10 1007 3 540 61064 2_27---，https://doi.org/ . / - - -14。Bozzano, M., Cimatti, A., Fernandes Pires, A., Jones, D., Kimberly, G., Petri, T., Robinson, R., Tonetta, S.: AIR6110 机轮制动系统的形式化设计与安全性分析。载于：Kroening, D., P˘as˘areanu, C.S.（编）计算机辅助验证。第 518–535 页。Springer International Publishing，Cham（2015）15. Bradley, A.R., Manna, Z.: 通过归纳反例的归纳泛化检验安全性。载于：FMCAD'07。第 173–180 页。IEEE计算机学会（2007）16. Bradley, A.: IC3 参考实现：一个简短、简单且颇具竞争力的 IC3 实现（2015），https://github.com/arbrad/IC3ref 17. Brayton, R., Mishchenko, A.: ABC：一种具有学术与工业级实力的验证工具。载于：CAV'10。第 24–40 页。Springer（2010）

18. Brzozowski, J.A.：确定事件的规范正则表达式与最小状态图。载于：数学自动机理论研讨会论文集。第 529–561 页（1962）19. B¨uchi, J.R.：关于受限二阶算术中的一种判定方法。载于：1960 年逻辑、方法与科学哲学国际大会论文集。斯坦福大学出版社，斯坦福（1962）20. Cavada, R., Cimatti, A., Dorigatti, M., Griggio, A., Mariotti, A., Micheli, A., Mover, S., Roveri, M., Tonetta, S.：nuxmv 符号模型检验器。载于：计算机辅助验证。第 334–342 页。Springer International Publishing，Cham（2014）21. C´ec´e, G.：高效仿真算法系列的基础。载于：第 32 届 ACM/IEEE 计算机逻辑年会，LICS 2017，冰岛雷克雅未克，2017 年 6 月 20–23 日。第 1–12 页。IEEE计算机学会（2017）。https://doi.org/10.1109/LICS.2017 8005069 22 8005069, 10 1109 2017 . https://doi.org/ . /LICS. . . Chen, T., Chen, Y., Hague, M., Lin, A.W., Wu, Z.：带有 replaceall 函数的字符串约束中哪些问题是可判定的。ACM 程序设计语言论文集 **2**（POPL），3:1–3:29（2018）。https: //doi.org/ 10 1145 3158091, 10 1145 3158091 . / https://doi.org/ . / . 23 Chen, T., Hague, M., Lin, A.W., Rummer, P., Wu, Z.：具有复杂操作的字符串处理程序路径可行性的判定过程。ACM 程序设计语言论文集 **3**（POPL），49:1–49:30（2019）。https://doi.org/10.1145/3290362, https://doi.org/ 10.1145/3290362 24. Cox, A.：正则表达式模型检验。URL：https://mosca19.github.io/ slides/cox.pdf（2019 19 25 ），发表于 MOSCA’ 。Cox, A., Leasure, J.：正则语言约束的模型检验。CoRR **abs/1708.09073**（2017）26. D’Anthoni, L.：symbolicautomata，https://github.com/lorisdanto/symbolicautomata 27. D’Antoni, L., Kincaid, Z., Wang, F.：符号交替有限自动机的符号判定过程。理论计算机科学电子札记 **336**，79–99（2018）。https://doi.org/https://doi.org/10.1016/j.entcs.2018.03.017, https: //www.sciencedirect.com/science/article/pii/S1571066118300203，第 33 届程序设计语义数学基础会议（MFPS XXXIII）28. D’Antoni, L., argus Veanes：符号树自动机的最小化。载于：LICS’16。第 873– 882 页。ACM 计算逻辑汇刊（2016）29. D’Antoni, L., Veanes, M.：符号自动机的最小化。载于：第 41 届 ACM SIGPLAN-SIGACT 程序设计语言原理研讨会论文集。第 541–553 页。POPL ’14，美国计算机协会，纽约，NY，美国（2014）。https://doi.org/10.1145/2535838.2535849, https://doi.org/10.1145/2535838. 2535849 30. D’Antoni, L., Veanes, M.：符号自动机与转换器的力量。载于：计算机辅助验证。第 47–67 页。Springer International Publishing，Cham（2017）31. De Giacomo, G., Vardi, M.Y.：有限轨迹上的线性时序逻辑与线性动态逻辑。载于：IJCAI’13。第 854–860 页。ACM（2013）32. De Wulf, M., Doyen, L., Maquet, N., Raskin, J.F.：Alaska。载于：Cha, S.S., Choi, J.Y., Kim, M., Lee, I., Viswanathan, M.（编）：验证与分析自动化技术。第 240–245 页。Springer Berlin Heidelberg，柏林，海德堡（2008）33. Doyen, L., Raskin, J.：有限自动机的反链算法。载于：系统构造与分析工具和算法，第 16 届国际会议，TACAS 2010，作为软件理论与实践联合欧洲会议 ETAPS 2010 的一部分举行，塞浦路斯帕福斯，2010 年 3 月 20–28 日。论文集。计算机科学讲义，第 6015 卷，第 2–22 页。Springer（2010）。https://doi.org/10.1007/978-3-642-12002-2_2, https://doi.org/10.1007/978-3-642-12002-2_2

34. E´en, N., S¨orensson, N.：可扩展的 SAT 求解器。载于：可满足性测试理论与应用，第 6 届国际会议，SAT 2003。意大利圣玛格丽塔利古雷，2003 年 5 月 5–8 日，修订论文选。计算机科学讲义，第 2919 卷，第 502– 518 页。Springer（2003）。 https://doi.org/10.1007/978-3-540-24605-3_37，https:// doi.org/ 10 1007 978 3 540 24605 3_37 。/ - - - - 。Fu, C., Deng, Y., Jansen, D.N., 35 Zhang, L.：非确定有限自动机的等价性检验。载于：可信软件工程：理论、工具与应用——第三届国际研讨会 SETTA 2017，中国长沙，2017 年 10 月 23– 25 日，论文集。计算机科学讲义，第 10606 卷，第 216–231 页。Springer（2017）。 https://doi.org/10.1007/978-3-319-69483-2_13，https://doi. org/ 10 1007 978 3 319 69483 2_13 。/ - - - - 。Fu, C., Deng, Y., Jansen, D.N., Zhang, 36 L.：非确定有限自动机的等价性检验。载于：可信软件工程：理论、工具与应用。第 216–231 页。Springer International Publishing，Cham（2017）37. Gange, G., Navas, J.A., Stuckey, P.J., Søndergaard, H., Schachte, P.：使用插值对正则语言约束进行无界模型检验。载于：TACAS 2013。计算机科学讲义，第 7795 卷，第 277–291 页。Springer（2013）。 https://doi.org/10. 1007 978 3 642 36742 7_20 10 1007 978 3 642 36742 7_20 / - - - - , 38 https://doi.org/ 。/ - - - - 。Ganty, P., Maquet, N., Raskin, J.：针对交替自动机的定点引导抽象精化。理论计算机科学**411**（38-39），3444–3459（2010）39. Gario, M., Cimatti, A., Mattarei, C., Tonetta, S., Rozier, K.Y.：大规模模型检验：自动化空中交通管制设计空间探索。载于：Chaudhuri, S., Farzan, A.（编）：计算机辅助验证。第 3–22 页。Springer International Publishing，Cham（2016）40. Harding, A.：具有 LTL 胜利条件的博弈符号策略综合。博士学位论文，伯明翰大学（2005）41. Henriksen, J.G., Jensen, J.L., Jørgensen, M.E., Klarlund, N., Paige, R., Rauhe, T., Sandholm, A.：Mona：实践中的单子二阶逻辑。载于：TACAS '95。LNCS，第 1019 卷，第 89–110 页。Springer（1995）42. Henzinger, M.R., Henzinger, T.A., Kopke, P.W.：有限图与无限图上的仿真计算。载于：第 36 届计算机科学基础年会，美国威斯康星州密尔沃基，1995 年 10 月 23–25 日。第 453–462 页。IEEE计算机学会（1995）。https://doi.org/10.1109/SFCS.1995.492576, https://doi.org/10.1109/ SFCS.1995.492576 43. Hoder, K., Bjørner, N.：广义性质导向可达性。载于：SAT'12。LNCS，第 7317 卷，第 157–171 页。Springer（2012）44. Hol´ík, L., Jank˚u, P., Lin, A.W., R¨ummer, P., Vojnar, T.：高效求解带有连接和转换器的字符串约束。ACM 程序设计语言论文集 **2**（POPL）（2018）45. Hol´ík, L., Leng´al, O., S´ı˘c, J., Veanes, M., Vojnar, T.：符号自动机的仿真算法。载于：Lahiri, S.K., Wang, C.（编）：验证与分析自动化技术。第 109–125 页。Springer International Publishing，Cham（2018）46. Hol´ík, L., Leng´al, O., Sim´acek, J., Vojnar, T.：显式与半符号树自动机上的高效包含性检验。载于：Bultan, T., Hsiung, P.（编）：验证与分析自动化技术，第 9 届国际研讨会，ATVA 2011，中国台北，2011 年 10 月 11–14 日。论文集。计算机科学讲义，第 6996 卷，第 243–258 页。Springer （ 2011 ）。https://doi.org/10.1007/978-3-642-24372-1_18，https:// doi.org/ 10 1007 978 3 642 24372 1_18 。/ - - - - 。Holık, L., ˇSima˘cek, J.：47 优化 LTS 仿真算法。Computing and Informatics**2010**9733 48 7 1337 1348 2010（），–（），https://www.fit.vut.cz/research/publication/ 。Hooimeijer, P., Weimer, W.：正则语言上的子集约束判定过程。载于：第 30 届 ACM SIGPLAN 编程语言设计与实现会议论文集。第 188–198 页。PLDI '09，美国计算机协会

美国计算机协会，纽约州纽约市（2009）。
https://doi.org/10.1145/1542476.1542498, https://doi.org/10 1542498 49 1145 1542476 . / . . Hopcroft, J.E.：一种用于最小化有限自动机状态数的 n log n 算法。技术报告，斯坦福，加利福尼亚州，美国（1971）50. Huffman, D.：时序开关电路的综合。富兰克林学会杂志 **257**(3), 161–190（1954）。https://doi.org/https://doi.org/10.1016/0016-0032(54)90574-8, https://www.sciencedirect.com/science/article/pii/0016003254905748 51. Ilie, L., Navarro, G., Yu, S.：关于 NFA 约简，第 112–124 页。Springer Berlin Heidelberg，柏林，海德堡（2004）。
https://doi.org/10.1007/978-3-540-27812-2_11, https://doi.org/10 1007 978 3 540 27812 2_11 . / - - - - . Iosif, R., Xu, X.：交替
52 数据自动机空性检查的抽象精化。载于：TACAS'18。第 93–111 页。Springer（2018）53. Leng´al, O., Sim´acek, J., Vojnar, T.：VATA：用于高效操作非确定树自动机的库。载于：系统构造与分析工具和算法——第 18 届国际会议 TACAS 2012，作为欧洲软件理论与实践联合会议 ETAPS 2012 的一部分，于 2012 年 3 月 24 日至 4 月 1 日在爱沙尼亚塔林举行。会议论文集。计算机科学讲义，卷 7214，第 79–94 页。Springer（2012）。
https://doi.org/10.1007/978-3-642-28756-5_7, https:// doi.org/10 1007 978 3 642 28756 5_7 . / - - - - . Li, J., Pu, G., Zhang, Y., Vardi,
54 M.Y., Rozier, K.Y.：基于 SAT 的显式 LTLf 可满足性检查。人工智能 **289**, 103369（2020）。https://doi.org/https://doi. org/10.1016/j.artint.2020.103369, https://www.sciencedirect.com/science/ article/pii/S0004370220301193 55. Lutterkort, D.：libfa，https://augeas.net/libfa/ 56. McMillan, K.L.：插值与基于 SAT 的模型检测。载于：计算机辅助验证。第 1–13 页。Springer Berlin Heidelberg，柏林，海德堡（2003）57. McMillan, K.L.：基于插值项的惰性抽象。载于：CAV'06。LNCS，卷 4144，第 123–136 页。Springer（2006）58. Moore, E.F.：关于时序机的思想实验，第 129–154 页。普林斯顿大学出版社，普林斯顿（1956）。
https://doi.org/doi:10.1515/9781400882618-006, https://doi.org/10 006 59 1515 9781400882618 3. / - . de Moura, L., Bjørner, N.：Z：一种高效的 SMT 求解器。载于：系统构造与分析工具和算法。第 337–340 页。Springer Berlin Heidelberg，柏林，海德堡（2008）60. N¨otzli, A., Reynolds, A., Barbosa, H., Barrett, C., Tinelli, C.：字符串求解器中更快的冲突处理和更惰性的约简。载于：计算机辅助验证。第 205–226 页。Springer International Publishing，Cham（2022）61. Paige, R., Tarjan, R.E.：三种划分精化算法。SIAM 计算杂志 **16**(6), 973–989（1987）。
https://doi.org/10.1137/0216062, https://doi.org/ 10.1137/0216062 62. Ranzato, F., Tapparo, F.：一种基于抽象解释的高效模拟算法。信息与计算**208**, 1–22（2010）63. RegExLib.com：互联网首个正则表达式库。http://regexlib.com/ 64. Rozier, K.Y., Vardi, M.Y.：LTL 可满足性检查。载于：SPIN'07。第 149–167 页。Springer（2007）65. Stanford, C., Veanes, M., Bjørner, N.：用于高效求解扩展正则表达式约束的符号布尔导数。载于：第 42 届 ACM SIGPLAN 编程语言设计与实现国际会议论文集。第 620–635 页。PLDI 2021，美国计算机协会，纽约，美国（2021）。https://doi.org/10.1145/3453483.3454066, https://doi.org/10.1145/3453483.3454066

66. Stanford, C., Veanes, M., Bjørner, N.S.：用于高效求解扩展正则表达式约束的符号布尔导数。载于：PLDI '21。第 620–635 页。ACM（2021）。https:// doi.org/10 3454066 67 1145 3453483 3454066, 10 1145 3453483 . / . https://doi.org/ . / . . Tabakov, D., Vardi, M.Y.：经典自动机构造的实验评估。载于：编程、人工智能与推理逻辑。第 396–411 页。Springer Berlin Heidelberg，柏林，海德堡（2005）68. Valmari, A.：一种在 o(m log n) 时间内实现的简单双模拟最小化方法。Fundam. Informaticae **105**(3), 319–339（2010）。https://doi.org/10.3233/FI-2010-369, https://doi.org/ 10.3233/FI-2010-369 69. Vargovˇc´ık, P., Hol´ık, L.：简化交替自动机以进行空性检测。载于：编程语言与系统。第 243–264 页。Springer International Publishing，Cham（2021）70. Veanes, M.：https://github.com/AutomataDotNet 71. Veanes, M., de Halleux, P., Tillmann, N.：Rex：符号正则表达式浏览器。载于：第三届软件测试、验证与确认国际会议 ICST 2010，法国巴黎，2010 年 4 月 7–9 日。第 498–507 页。IEEE计算机学会（2010）。https://doi.org/10.1109/ICST.2010.15, https://doi.org/10.1109/ICST.2010.15 72. Wang, H., Tsai, T., Lin, C., Yu, F., Jiang, J.R.：通过带有逻辑电路表示的自动机操作进行字符串分析。载于：CAV'16。LNCS，卷 9779，第 241–260 页。Springer（2016）73. Wolper, P., Boigelot, B.：Presburger 算术约束的自动机理论方法（扩展摘要）。载于：Mycroft, A.（编）静态分析：第二届国际研讨会 SAS'95，英国格拉斯哥，1995 年 9 月 25–27 日，会议论文集。计算机科学讲义，卷 983，第 21–32 页。Springer（1995）。https://doi.org/10.1007/3-540-60360-3_30, https://doi.org/10.1007/3-540-60360-3_3074. Wulf, M.D., Doyen, L., Henzinger, T.A., Raskin, J.：反链：一种用于检查有限自动机全性的全新算法。载于：CAV'06。LNCS，卷 4144，第 17–30 页。Springer（2006）

最佳工具的一对一比较

此外，对于图 1 中展示的每个基准测试，我们根据超时率或平均用时选出了表现最好的三种工具。我们在图 3 中以散点图的形式展示了它们的比较。每组三个工具以点的颜色加以区分，每组三个工具的位置对应其基准测试在图 1 中的位置。

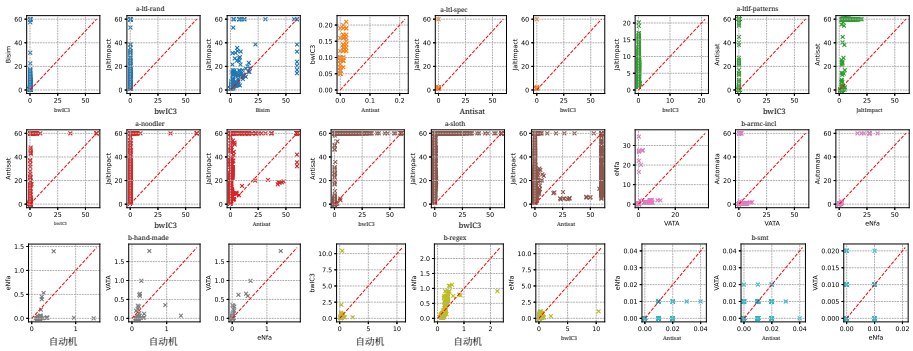


图 3. 按基准测试对最佳工具进行一对一比较的散点图。