构建简洁的　　通过约束 Tsetlin 机器子句大小构建逻辑模式

K. Darshana Abeyrathna[1] , **Ahmed Abdulrahem Othman Abouzeid**[2] , **Bimal Bhattarai**[2] , **Charul Giri**[2] , **Sondre Glimsdal**[2] , **Ole-Christoffer Granmo**[2] , **Lei Jiao**[2] , **Rupsa Saha**[2] , **Jivitesh Sharma**[2] , **Svein Anders Tunheim**[2] 和 **Xuan Zhang**[3*]

[2]人工智能研究中心（CAIR），阿格德尔大学，克里斯蒂安桑，挪威[3]NORCE 挪威研究中心，[1]DNV 挪威 ole.granmo@uia.no

arXiv:2301.08190v1 [cs.LG] 19 Jan 2023

摘要

Tsetlin 机器（TM）是一种基于逻辑的机器学习方法，具有透明且适用于硬件的关键优势。随着越来越多的应用中 Tsetlin 机器达到或超越深度学习的准确率，大型子句池往往会产生包含大量文字的子句（长子句）。因此，它们的可解释性会降低。此外，较长的子句会增加硬件中子句逻辑的开关活动，从而消耗更多功耗。本文介绍了一种新型 TM 学习变体——子句大小受约束的 Tsetlin 机器（CSC-Tsetlin 机器）——用户可以为子句大小设定软约束。一旦子句包含的文字数量超过约束允许的上限，它便开始剔除文字。因此，超大子句只会暂态出现。为评估 CSC-TM，我们在表格数据、自然语言文本、图像和棋盘游戏上进行了分类、聚类和回归实验。结果表明，CSC-TM 在文字数量减少至原来的 1/80 时仍能保持准确率。事实上，在 TREC、IMDb 和 BBC Sports 数据集上，随着子句变短，准确率反而会提高。准确率达到峰值后，随着子句大小趋近于单个文字，准确率会平缓下降。最后，我们分析了 CSC-TM 的功耗，并推导出新的收敛性质。

1 引言

Tsetlin 机 [Granmo, 2018] 是一种新颖的机器学习方法，其中多组 Tsetlin 自动机 (Tsetlin 自动机) [Tsetlin, 1961] 会生成由合取子句（与规则）构成的逻辑（布尔）表达式。与深度神经网络的黑箱特性不同，Tsetlin 机器本质上具有可解释性。事实上，它们生成基于稀疏析取范式的模型，相较之下更易于人类理解 [Valiant, 1984]。此外，逻辑表示与基于自动机的学习相结合，使 Tsetlin 机器天然适用于硬件实现，从而降低能耗 [Wheeldon 等, 2020]。

Tsetlin 机器现已支持多种架构，包括分类 [Granmo, 2018]，卷积 [Granmo 等，2019]，回归 [Abeyrathna 等，2020年b]，确定性 [Abeyrathna 等，2020年a]，加权 [Abeyrathna 等，2020年c]，自编码器 [Bhattarai 等，2023]，上下文老虎机 [Seraj 等，2022]，关系型 [Saha 等，2022]，以及多输入多输出 [Glimsdal 和 Granmo，2021] 架构。子句学习的独立特性支持高效的基于 GPU 的并行化，在子句数量合理时可实现近乎恒定的时间扩展 [Abeyrathna 等，2021]。若干方案改进了原始 Tsetlin 机器学习和推理，例如 Drop clause [Sharma 等，2023] 以及聚焦负采样 [Glimsdal 等，2022]。这些 Tsetlin 机器的进展推动了众多应用的发展：关键词检测 [Lei 等，2021],基于方面的情感分析 [Yadav 等，2021年b]，新颖性检测 [Bhattarai 等，2022年b]，语义关系分析 [Saha 等，2021]，文本分类 [Yadav 等，2021年a；Bhattarai 等，2022年a；Yadav 等，2022]，游戏对弈 [Giri 等，2022]，无电池传感 [Bakar 等，2022年b；Bakar 等，2022年a]，推荐系统 [Borgersen 等，2022]，以及词嵌入 [Bhattarai 等，2023]。

尽管 Tsetlin 机器在越来越多的应用中，其准确率已达到或超过深度学习，但规模较大的子句池往往会产生更长的子句，其中包含许多文字（输入特征及其否定形式）。因此，这些子句的可解释性会降低。此外，较长的子句需要更多内存，并会增加硬件中的开关活动，从而消耗更多功耗。目前还没有直接的方法来控制所学习子句的大小。挑战在于如何协调分散的 Tsetlin 自动机；每个自动机都独立学习是否在每个子句中包含特定文字。由于 Tsetlin 自动机旨在寻找频繁出现的判别模式，子句的长度可能会变得任意长。简而言之，现有学习方案在所生成子句的大小方面似乎效率低下。

本文介绍了一种新颖的 TM 学习变体——子句大小受限 Tsetlin 机器（CSC-Tsetlin 机器）——它允许对子句大小设置软约束。CSC-TM 修订了将文字纳入子句的 Tsetlin 自动机反馈策略。具体而言，一旦子句长度超过预设约束，新策略便会抑制继续纳入文字。相反，Tsetlin 自动机立即开始通过强化“排除”动作，将文字从超出约束的子句中移除。

*作者按姓氏字母顺序排列。

因此，超大子句只会暂态出现。除此之外，TM 反馈方案保持不变。我们的方法最显著的特点是，最终保留在子句中的有限文字集合仍具有很强的判别能力。即使子句大小受到显著限制，与其他 TM 变体相比，CSC-TM 的性能也不会受到影响。

论文贡献： 本文的贡献可总结如下：

- 我们提出 CSC-TM，通过引入一种新的 Tsetlin 机器训练策略来约束子句大小。
- 我们证明，CSC-TM 确实能够将子句大小约束在明确的限值内，同时不影响分类、回归和聚类的准确率。
- 通过多个示例，我们展示了较短的子句更具可解释性。
- 我们从理论上证明，经过适当配置后，CSC-TM 可以收敛到预期的基本运算符。
- 我们阐述了约束子句长度如何有益于嵌入式硬件解决方案的功耗：子句逻辑开关活动的减少可降低功耗。

2 使用受限子句长度训练 Tsetlin 机

本节概述原始TM与 CSC-TM 之间的区别。对于不熟悉 TM学习的读者，详细说明可见附录 1。

TM 将向量 $\mathbf{X} = [x_1, \ldots, x_o]$ 作为输入，其中包含命题（布尔）特征，并将其分类为两个类别之一：$y = 0$ 或 $y = 1$。对这些特征取反会生成一组文字 L，由这些特征及其取反对应项组成：$L = \{x_1, \ldots, x_o, \neg x_1, \ldots, \neg x_o\}$。

TM 使用合取子句表示子模式。子句数量由用户设定的参数 n决定。对于二分类器[1]，，一半子句具有正极性（+）。另一半具有负极性（–）。每个子句 C_j^p，$j \in \{1, 2, \ldots, n/2\}$，$p \in \{-, +\}$，随后变为：

$$C_j^p(\mathbf{X}) = \bigwedge_{l_k \in L_j^p} l_k. \tag{1}$$

此处，j 是子句的索引，p 是其极性，而 L_j^p 是文字 L、$L_j^p \subseteq L$的一个子集。例如，子句$C_1^+(\mathbf{X}) = \neg x_1 \wedge x_2$ 的索引为 1，极性为 +，由文字 $L_1^+ = \{\neg x_1, x_2\}$组成。因此，当 $x_1 = 0$且 $x_2 = 1$时，该子句输出 1，否则输出 0。

子句输出通过求和，并使用单位阶跃函数 $u(v) = 1$ **如果** $v \geq 0$ **否则** 0 进行阈值处理，从而组合成分类决策：

$$\hat{y} = u\left(\sum_{j=1}^{n/2} C_j^+(\mathbf{X}) - \sum_{j=1}^{n/2} C_j^-(\mathbf{X})\right). \tag{2}$$

也就是说，分类基于多数表决进行，正子句为 $y = 1$ 投票负子句则为 $y = 0$ 投票。分类器 $\hat{y} = u((x_1 \wedge \neg x_2) + (\neg x_1 \wedge x_2) - (x_1 \wedge x_2) - (\neg x_1 \wedge \neg x_2))$，例如，捕获了异或关系。

在训练过程中，由一支由 Tsetlin 自动机构成的专门团队为每个子句 C_j^p 进行组合。子句 C_j^p 中的每个 Tsetlin 自动机都会决定在子句中包含还是排除 特定文字 l_k 。Tsetlin 自动机根据收到的奖励、不动作和惩罚形式的反馈作出决定。TM 学习有两种反馈类型：I型反馈和 II 型反馈。I型反馈会促成频繁模式的形成，从而抑制假阴性分类。另一方面，II 型反馈会增强模式的区分能力，以抵消假阳性分类。

原始TM 与 CSC-TM 的区别在于 I 型反馈。两种方案的 II 型反馈保持不变。表 1展示了 CSC-TM 如何修改 I 型反馈以约束子句大小。修改部分以红色突出显示。如图所示，我们增加了一个额外条件，用于触发最左侧的两列反馈。这两列通过以概率 $\frac{s-1}{s}$ 强化包含值为“1”的文字，并以概率 $\frac{1}{s}$ 强化排除值为“0”的文字，使子句模仿频繁模式。另一方面，最右侧的两列只会强化排除文字。

CSC-TM 要求子句大小 $\|C_j^p(\mathbf{X})\|$ 满足约束 $\mathbf{b}$，才能使用最左侧的两列。因此，一旦子句中的文字数量超过约束$\mathbf{b}$，就只会强化排除 动作。原因是此时所有 Tsetlin 自动机的反馈都来自最右侧的两列，这两列以概率 $\frac{1}{s}$ 分别奖励排除 并惩罚包含 。因此，子句过大时会开始剔除文字，这意味着过大的子句只会暂态出现。

$C_j^p(\mathbf{X}) \wedge (\|C_j^p(\mathbf{X})\| \leq \mathbf{b})$		1		0	
	$x_k/\neg x_k$	1	0	1	0
Tsetlin 自动机：**包含** **文字**	P(奖励)	$\frac{s-1}{s}$	NA	0	0
	P(不动作)	$\frac{1}{s}$	NA	$\frac{s-1}{s}$	$\frac{s-1}{s}$
	P(惩罚)	0	NA	$\frac{1}{s}$	$\frac{1}{s}$
Tsetlin 自动机：**排除** **文字**	P(奖励)	0	$\frac{1}{s}$	$\frac{1}{s}$	$\frac{1}{s}$
	P(不动作)	$\frac{1}{s}$	$\frac{s-1}{s}$	$\frac{s-1}{s}$	$\frac{s-1}{s}$
	P(惩罚)	$\frac{s-1}{s}$	0	0	0

表1：CSC-TM 的 I 型反馈。该反馈适用于单个 Tsetlin 自动机，用于决定是否将给定文字 $x_k/\neg x_k$ 包含到 C_j^p 中。NA 表示不适用。s 是一个大于 1 的超参数。

在以下区段中，我们分析 CSC-TM 对收敛的影响。我们还将进一步研究限制子句大小对分类、回归和聚类准确率的影响。最后，我们讨论其对硬件复杂度和能耗的影响。

3 收敛性分析

在此，我们分析 CSC-TM 使用两种基本运算符时的收敛特性。通过研究异或和或

[1]一个多类分类器的每类子句数为 n。

x_1	x_2	输出
0	0	0
1	1	0
0	1	1

表 2：“异或”情况下的子模式。

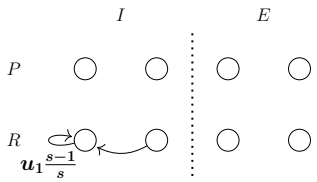
运算符，我们推测[2]，如果子句中的文字数量足以表示一个子模式（或一组子模式），Tsetlin 机就能学会预期的子模式（或一组子模式）。然而，如果表示某个子模式所需的文字数量超过文字预算，该子模式就无法学会。

异或运算符

这里我们研究仅给定一个文字时异或运算符的收敛情况，即 $(\|C_j^i(\mathbf{X})\| = \mathbf{1})$。接着可以证明，如果预算不足，就无法学到该子模式。显然，异或中的子模式彼此互斥，一个文字无法完整捕获异或的任何子模式。接下来，我们将展示 TM 对异或训练样本的反应。

正如 [Jiao 等人，Aug 2022], 已经证明的那样，当给定表 2 中的训练样本时，原始TM 几乎必然收敛到预期的子模式，即$C_j^i = \neg x_1 \wedge x_2$。然而，此外若给定 $(\|C_j^i(\mathbf{X})\| = \mathbf{1})$，系统唯一的吸收态，即 $C_j^i = \neg x_1 \wedge x_2$，便会消失，使系统变为常返。具体而言，对于原始 TM，根据 [Jiao 等人，Aug 2022], ，当向系统提供针对 $x_1 = 0, x_2 = 1, y = 1$ 的训练样本，且 TA_1^3=排除、TA_2^3=包含、TA_4^3=排除时，TA_3^3会发生如下状态转移[3]。

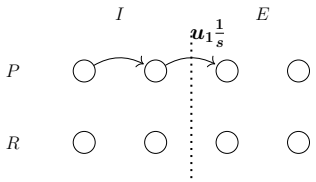
条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, TA_4^3=排除。因此，对于文字 $x_2 = 1$, $C_3 = \neg x_1 \wedge x_2 = 1$，我们有 I型反馈。



这里，TA_1^3 的上标是子句索引，下标是 TA 索引。TA_1^3 有两个动作，即包含 或排除 x_1。同样，TA_2^3 对应包含或排除 $\neg x_1$。TA_3^3 和 TA_4^3 分别决定 x_2和 $\neg x_2$的行为。这里， P 和 R 分别表示惩罚和奖励，而 I 和 E 分别表示包含 和排除。 u_1 是 $[0, 1]$中的一个常数。

显然，对于原始TM，新的训练样本 $x_1 = 0$、$x_2 = 1$、 $y = 1$ 将促使 Tsetlin 自动机 $\frac{3}{3}$ 被包含，从而强化 C_3 采用 $\neg x_1 \wedge x_2$ 形式。然而，当约束 $(\|C_j^i(\mathbf{X})\| = \mathbf{1})$ 被额外给出时，Tsetlin 自动机 $\frac{3}{3}$ 的状态转移变为：

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, TA_4^3=排除。因此，对于文字 $x_2 = 1$, $C_3 = \neg x_1 \wedge x_2 \wedge 0 = 0$，我们有 I 型反馈。



上述更改将使唯一的吸收态，即$C_3 = \neg x_1 \wedge x_2$，消失。可以理解的是，在一个文字已被包含的情况下，系统不会鼓励包含更多文字。因此，当子句长度为 1 时，TM 几乎肯定无法捕获 XOR 中的任何子模式。上述分析也证实，新添加的长度约束按预期发挥作用，即当文字预算达到上限后，它确实会抑制包含更多文字。完整证明见附录2。

或运算符

现在我们来研究或运算符，旨在说明：当子句中的文字预算足以表示一个子模式（或一组子模式）时，TM 能够学习预期的子模式（或预期的一组子模式）。在研究或运算符的收敛性之前，我们先回顾一下它的性质。有三个子模式可以触发正输出，即 $(x_1 = 1, x_2 = 1)$、$(x_1 = 0, x_2 = 1)$ 和 $(x_1 = 1, x_2 = 0)$。要明确（或分别）表示每个子模式，需要两个文字。不过，两个子子句也可以用一个文字共同表示。显然， $C = x_1$ 可以同时覆盖 $(x_1 = 1, x_2 = 1)$ 和 $(x_1 = 1, x_2 = 0)$，而 $C = x_2$ 可以同时覆盖 $(x_1 = 1, x_2 = 1)$ 和 $(x_1 = 0, x_2 = 1)$。这使 TM 有可能在超参数[4] T的配合下，通过仅含一个文字的子句学习预期的或运算符。

根据 [Jiao 等人的分析， 2021], 我们知道，如果 $T = \lfloor \frac{m}{2} \rfloor$，且当 T 子句学习 x_1 而另一些T 子句学习 x_2时，系统会进入吸收态。该吸收态能够学习预期的或运算符，也符合 CSC-TM 的要求，这表明在文字长度预算为 1 的情况下，CSC-TM 可能学习到或运算符。接下来，我们将说明，或运算符的其他吸收态（即子句需要一个以上文字的状态）不会受到鼓励，因为新加入了长度约束。

与异或情形的分析类似，当我们重新审视包含 2 个文字的吸收状态的转移时，我们意识到这些吸收状态不再具有吸收性。更具体地说，I 型反馈会促使已包含的文字被排除。例如，在原始TM中，以下子模式：

$$P(y = 1 | x_1 = 1, x_2 = 1) = 1, \quad (3)$$
$$P(y = 0 | x_1 = 0, x_2 = 0) = 1,$$

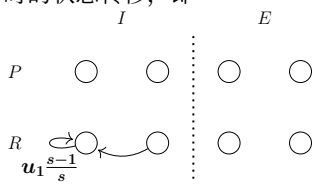
[2]这里的分析仅限于异或和或运算符，而非一般情况。因此，我们使用“猜想”一词。尽管如此，该分析仍能提供有用的见解。

[3] 这是情况 1 在第 3.2.1 小节中的第 2 次状态转移，见 [Jiao 等人，8月 2022]。状态转移图是根据系统的当前状态和输入的训练样本推导出来的。详情请参阅 [Jiao等人，8月 2022]。

[4]超参数 T 用于引导不同子句学习不同的子模式。详情请参见 [Jiao 等人，Aug 2022]。

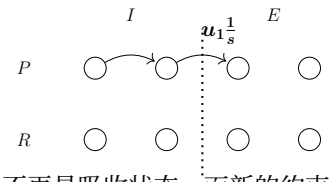
当 TA_3 的当前动作是 包含，且 $TA_1 =$ 的当前动作是 包含，而 $TA_2 =$ 的当前动作是 排除 时的状态转移，即

条件： $x_1 = 1, x_2 = 1, y = 1$, Tsetlin 自动机 $_4 =$排除。因此，I 型，$x_2 = 1, C = x_1 \wedge x_2 = 1$，



在 CSC-TM 中被替换为

Condition: $x_1 = 1, x_2 = 1, y = 1$, $TA_4 =$ Exclude.
Thus, Type I, $x_2 = 1, C = x_1 \wedge x_2 \wedge 0 = 0$.



显然，状态（即 $x_1 \wedge x_2$）不再是吸收状态，而新的约束会促使具有包含文字的 Tsetlin 自动机（此处为 TA_3,）转向“排除”。类似情况也适用于 $(x_1 = 1, x_2 = 0)$ 和 $(x_1 = 0, x_2 = 1)$。基于上述分析，我们可以得出结论：CSC-TM 仍然能够学习或运算符，但只能使用包含 1 个文字的子句。

请注意，尽管子句的长度受到约束，但子句的长度仍可能超过预算。首先，当系统被阻塞时，子句长度可能会超过预算。考虑或运算符的一个极端情况： T 子句具有 x_1，而 $T-1$ 子句遵循 x_2。在这种情况下，由于随机性，一个子句可能会基于单个训练样本变成 $\neg x_1 \wedge x_2$。此时，系统被 T 阻塞，之后将不再更新。尽管这种情况可能发生，但其概率非常低。原因是，这要求两个 Tsetlin 自动机恰好都处于 排除 侧的边界状态，而且必须同时有一个训练样本触发这两个 Tsetlin 自动机向 包含 侧转移。

超出预算的另一个原因是，II 型反馈不受约束，可能生成包含多于预算数量文字的子句。在这种情况下，包含的文字所对应的 Tsetlin 自动机都将处于中间状态，而这些文字会快速地进出子句，直到训练停止。通过这种方式，II 型反馈确保所有文字都得到探索。在文字探索期间，尽管子句长度可能很大，但在找到准确的文字配置之前，准确率会一直很低。因此，由于 II 型反馈而处于边界的包含文字不一定会对分类产生积极作用。当文字预算较低时，对图像处理的数值结果可以观察到这一点（表3中展示了 CIFAR-2 和带卷积的 MNIST 的结果）。具体而言，卷积需要更多文字，以便同时存储子句位置。文字过少时，子句的精度会受到影响，从而触发大量 II 型反馈。

由于文字预算紧张，子句将无法稳定下来。因此，包含的文字对应的 Tsetlin 自动机将无法深入其状态空间的包含 一侧。结果，II 型反馈将持续尝试新的候选文字，却无法找到足够稀疏且准确率高的配置。

4 实证结果

本区段中，我们将考察 CSC-TM 的性能，重点关注准确率和可解释性。为此，我们将评估其在自然语言、图像、棋盘游戏和表格数据等多种数据集上的分类、回归和聚类性能。实验采用 CSC-TM 的 CUDA 实现，并在 Intel Xeon Platinum 8168 CPU（2.70 GHz）和配备 Tesla V100 GPU 的 Nvidia DGX-2 上运行。我们将在相应的小区段中介绍各项任务的详细信息，并在表 3、4和 5中总结研究结果。

自然语言处理

我们首先在五个自然语言处理数据集上评估 CSC-TM：BBC Sports [Greene and Cunningham、 2006], R8 [Debole and Se- bastiani、 2005], TREC-6 [Chang et al.、 2002], SemEval 2010语义关系 [Hendrickx et al.，以及 ACL 互联网电影数据库（IMDb） [Maas et al. 2011]。从最大约束为 1 开始，我们逐步将文字约束提高到 64，并记录由此得到的准确率以及每个子句包含的文字平均数量。对于 BBC Sports，我们发现文字约束为 8 时，准确率最高，达到 99.1%。同样，TREC 和 R8 的最高准确率分别在文字数量为 32 和 all 时达到。我们还观察到，纳入所有文字会降低 BBC 和 TREC 的准确率。对于所有数据集，采用文字预算后，文字平均数量都显著下降。例如，BBC Sports 的文字约束为 32 时，每个子句平均包含 2.64 个文字，而不加约束时平均为 44.14 个。因此，子句长度大幅缩短，人们可以快速浏览子句并进行解读。其他数据集也呈现类似趋势。 [5]

子句	文字数量
$C_1 = [club \wedge football \wedge \neg six]$	3
$C_2 = [olympics \wedge title]$	2
⋮	⋮
$C_m = [goal \wedge winger]$	2
$C_1 = [manchester \wedge \neg australia \wedge \ldots \wedge \neg century]$	329
$C_2 = [\neg rugby \wedge \neg ship \wedge \ldots \wedge arsenal]$	331
⋮	⋮
$C_m = [barcelona \wedge matches \wedge \ldots \wedge injury]$	5

图1：BBC Sports 中“football”类别的子句可解释性，以及受约束子句大小的影响。红色高亮的行是来自 CLC-TM 的子句，黄色行则来自原始 TM。

以表3中 R8 的结果为例。请注意，如此严格的约束 4 仍几乎可以使准确率达到最大值。事实上，NLP 的整体结果表明，CSC-TM 能够帮助我们构建简洁且准确的命题规则，从而提升可理解性。改进后的可解释性

[5]对于 BBC Sports、TREC 和 R8 中的 TM 超参数，我们使用 8000 个子句、 T投票边际为 100，以及特异性 s 为 10。0

预算→	准确率（每个子句的文字数）							
	≤1	≤2	≤4	≤8	≤16	≤32	≤64	All
BBC Sports	98.65 (1.59)	98.65 (1.65)	98.65 (1.71)	99.1 (1.93)	98.2 (2.15)	98.2 (2.64)	98.2 (3.35)	94.14 (44.14)
TREC	91.8 (1.05)	91.6 (1.07)	92.4 (1.07)	90.2 (1.08)	90.4 (1.1)	93.2 (1.12)	90.6 (1.16)	85.8 (96.24)
R8	95.08 (1.09)	94.54 (1.13)	95.08 (1.14)	94.9 (1.19)	95.08 (1.26)	95.26 (1.41)	95.08 (1.7)	95.8 (21.84)
加利福尼亚住房 (5-箱)	59.25 (1.08)	62.28 (1.46)	64.2 (3.09)	65.02 (6.66)	65.24 (12.92)	65.26 (17.15)	65.29 (18.48)	65.79 (20.22)
SemEval	93.65 (1.23)	93.00 (1.26)	92.25 (1.32)	93.2 (1.41)	92.95 (1.62)	93.00 (2.88)	93.1 (5.67)	93.63 (142.97)
IMDb	81.58 (1.08)	81.51 (1.22)	81.28 (1.28)	82.01 (1.42)	83.44 (1.75)	85.67 (2.76)	87.73 (4.05)	84.23 (27.02)
CIFAR-2	69.82 (30.8)	79.65 (30.1)	87.01 (13.5)	91.21 (6.8)	93.23 (10.8)	93.99 (20.1)	94.24 (34.1)	94.18 (60.4)
MNIST	92.09 (1.0)	97.42 (1.4)	98.34 (3.0)	98.40 (5.8)	98.42 (10.8)	98.38 (19.9)	98.33 (33.0)	98.32 (47.7)
带卷积的 MNIST	40.94 (18.6)	60.55 (15.3)	95.93 (5.8)	99.22 (7.1)	99.33 (13.4)	99.29 (23.5)	99.30 (34.2)	99.28 (40.3)
能效 (回归)	5.65 (1.0)	2.44 (1.9)	1.05 (3.9)	0.86 (6.2)	0.78 (9.3)	0.66 (11.2)	0.63 (11.3)	0.59 (11.5)
Hex (强化学习)	67.59 (2.7)	74.69 (2.4)	77.79 (3.0)	79.49 (4.2)	81.23 (6.1)	81.60 (9.3)	82.17 (10.2)	81.43 (13.8)

表3：在文字预算分别为 1、2、4、8、16、32、64 以及全部文字（无约束）时，模型在多个数据集上的表现。除回归分析外，每个单元格报告 5 次独立运行中取得的最低最高准确率，括号内为 TM 使用的文字平均数量。对于能效，我们报告平均绝对误差，而非准确率。

预算→	准确率（每个子句的文字数）							
	≤1	≤2	≤4	≤8	≤16	≤32	≤64	All
MNIST（含 250 个子句）	88.20 (1.1)	93.02 (1.4)	95.96 (2.1)	96.65 (2.8)	96.92 (3.2)	97.03 (3.6)	97.06 (3.8)	97.04 (4.0)
MNIST（含 500 个子句）	89.05 (1.0)	94.28 (1.3)	96.82 (2.1)	97.50 (3.2)	97.67 (4.5)	97.74 (6.2)	97.78 (7.8)	97.77 (9.0)
MNIST，含 1000 个子句	90.22 (1.0)	95.26 (1.2)	97.67 (2.6)	98.03 (4.7)	98.10 (8.4)	98.14 (14.6)	98.09 (22.8)	98.05 (31.2)
MNIST，含 2000 个子句	90.88 (1.0)	96.23 (1.2)	98.01 (2.7)	98.26 (5.3)	98.26 (9.7)	98.33 (17.5)	98.22 (28.5)	98.2 (40.1)
MNIST（含4000个子句）	91.68 (1.0)	97.08 (1.3)	98.19 (2.9)	98.37 (5.6)	98.4 (10.5)	98.35 (19.1)	98.32 (31.4)	98.29 (45.2)
MNIST，含 8000 个子句	92.09 (1.0)	97.42 (1.4)	98.34 (3.0)	98.40 (5.8)	98.42 (10.8)	98.38 (19.9)	98.33 (33.0)	98.32 (47.7)

表 4：不同总子句数下在 MNIST 上的性能，每个子句的文字预算分别为 1、2、4、8、16、32、64 和全部（无约束）文字。每个字段报告 5 次独立运行中的最低最高准确率，后接方括号中 TM 使用的平均文字数。

如图 1所示，我们从 BBC Sports 的“足球”类别中提取了几个典型子句。受文字数约束的子句（红色）仅包含 2–3 个文字，并且与“足球”类别明显相关。而原始子句（黄色）则包含更多文字，并且大量依赖特征取反。

图像处理

我们在两个图像数据集上评估了所提方法：MNIST 和 CIFAR-2（CIFAR-10 的一个双类别变体，将车辆图像和动物图像归为两个独立类别）。[6]对于 MNIST，我们使用受子句长度约束的原始 Tsetlin 机和卷积 Tsetlin 机进行了实验。从表 3 可以看出，即使约束小至 8，仍能获得具有竞争力的准确率。具体而言，在 CIFAR-2、MNIST 和带卷积的 MNIST 上，最高准确率分别是在文字约束为 64、16和 16时取得的。还可以注意到，CSC-TM 平均而言，每个子句所含文字数远低于设定的约束值。例如，对于上述约束，每个子句的平均文字数分别为 34。1、10。8和 13。4。然而，在没有子句长度约束的情况下，平均包含文字数分别为 60。4、47。7和 40。3。最后请注意，对于带卷积的 MNIST 和 CIFAR-2，将文字预算设得过低（低于 4）会增加实际使用的平均文字数。这可以解释为 CSC-TM 无法找到准确率足够高的模式。因此，它在整个训练轮次中始终处于文字探索模式。总之，我们发现，使用 CLC-TM 可以在每个子句使用显著更少文字的情况下达到最高准确率。这使我们能够显著降低计算复杂度，并提高子句的可读性。

为了研究子句数量与文字约束之间的相互作用，我们现在测量同时改变子句数量和文字预算所产生的影响。从表4中可以看出，随着子句数量降至 250 且文字预算降至 1，准确率会逐渐下降。还可以注意到，子句数量越少，平均每个子句包含的文字也越少。我们认为这是因为可用子句较少时，每个子句必须降低专门化程度才能完成任务。然而，在考察如何达到最高准确率时，我们发现，1000 个子句所需的文字数多于 4000 个子句（32 比 16，均为平均每个子句的文字数）。

原因可能是，子句较少时需要更加具体，以作出补偿并维持准确率。随着子句数量增加，每个子句包含的文字会减少，并以集成的方式解决任务。总之，CSC-TM 能够在子句长度与子句数量之间实现细粒度的权衡。

自监督学习

对于自监督学习任务，我们评估子句文字预算对训练时间和可解释性的影响。在此，我们评估先前提出的标签批评 TM（LCTM） [Abouzeid 等人， 2022], 这是一种新颖的架构，无需了解真实标签即可自学习数据样本的标签。LCTM 架构运行于标准 CUDA TM 实现之上，首先随机初始化数据标签。随后，它在学习子模式及其相关标签的同时进行层次聚类。最终，学习到的子模式表示可解释簇，每个簇都与一个支持子句和一个区分子句相关联。因此，该方法具有可解释性，但仍可受益于更小的子句。

表 5 展示了在 MNIST 数据集的一个子集上，不同子句文字预算对应的实证结果。此处，LCTM 用于学习 MNIST 样本中与数字“One”和“Zero”相关的标签和子模式。结果反映了文字约束对训练时间和可解释性的影响。可解释性指标如下： 支持性可解释性 指经人工确认可识别的正极性子句所占的百分比。可参见图 2，其中展示了被认为可解释和不可解释的子句示例。同样，区分性可解释性 指经人工识别的负极性子句所占的百分比。

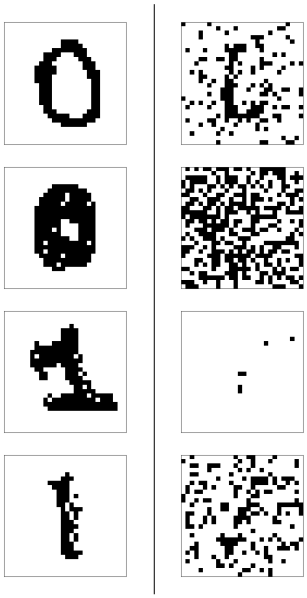


图 2：四个被认为人类可解释的聚类（好）和四个不可解释的聚类（差）的示例。

如表 5所示，当文字预算降至 1，200时，LCTM 的运行速度更快，生成的子句也更具可解释性。总之，限制子句大小既提高了可解释性，也提升了 LCTM 在所有评估中的学习速度。

回归

我们使用能效数据集，依据 [Abeyrathna 等， 2020b]评估回归性能。实验结果以平均绝对误差（MAE）报告。简而言之，表3 显示，MAE 会降低，

[6]在 MNIST 实验中，我们使用的 TM 超参数为：每类 8000 个子句、投票边界 T为 10000，特异性为 5。0。对于 CIFAR-2，子句数为 8000， T为 6000，s 为 10。0。

文字数 #	支持性可解释性 (%)	区分性可解释性 (%)	聚类数 #	加速
400	84.92 ± 5.21	32.17 ± 6.34	36 ± 7.04	1.1×
800	85.48 ± 7.80	33.02 ± 7.21	35.8± 6.76	1.3 ×
1,200	88.84 ± 2.71	39.05 ± 8.03	34.6 ± 9.54	1.3 ×
1,568（全部）	86.51 ± 1.33	33.63 ± 7.59	33.8 ± 7.34	1 ×

表 5：LCTM 在 MNIST 上采用不同文字预算时的表现。均值和标准差根据 5 次独立运行计算得出。

也就是说，随着文字预算从 1 增加到 64，性能会有所提升。同样，性能的下降是渐进的，因此可以在子句大小与 MAE 之间权衡。

棋盘游戏赢家预测

这里以六角棋为例，说明如何使用 CSC-TM 进行强化学习，任务是预测给定棋盘配置的赢家（价值）。为研究 CSC-TM 的预测准确率如何变化，我们比较了原始TM [Giri 等人， 2022] 与采用不同文字预算的 CSC-TM。实验设置的详细信息见 [Giri 等人， 2022]。表3 底行总结了准确率随文字预算的变化情况。我们观察到，文字预算为64 时，准确率达到最高值82。17% ，而使用全部文字时的准确率为81。43% 。不过需要注意的是，较小的文字预算会导致准确率相对较低。我们认为，这是因为六角棋棋盘配置的性质决定了，要准确描述它通常需要足够数量的棋子位置等信息。

5 硬件复杂度与能耗分析

TM 硬件加速器通常采用现场可编程门阵列（FPGAs）或专用集成电路（ASICs）实现。在大多数情况下，能耗主要来自数字电路的开关活动。不过需要注意，对于高性能工艺，晶体管漏电流导致的静态功耗最高可达总功耗的 30% [Dally 等人， 2016]。采用低功耗工艺实现时，静态功耗会低得多。

数字电路消耗的动态功耗 P与其负载电容 C、工作频率 f、电源电压V_S以及活动因子 α （每个时钟周期的状态转移次数）有关，其计算公式见式（4） [Dally等人， 2016],

$$P = 0.5 \times C \times V_S^2 \times f \times \alpha. \tag{4}$$

限制文字的数量会降低实现子句逻辑的若干门电路中的 α 值。只有处理包含的文字的门电路才会发生状态转移并消耗能量。因此，子句逻辑的能耗降低幅度大致可估算为$\mathbf{b}\, l_{ave}$，其中$\mathbf{b}$ 是子句大小约束，而l_{ave} 是不受长度约束时模型中的平均文字数。确切的节能效果取决于数据集。

经典方法 [Wheeldon 等人， 2020] 用于实现子句逻辑，如图3所示。此处，每个文字 l_1，……， l_{2o}通过或门，根据相应的包含信号 i_1，……，i_{2o}（低电平有效）将其包含或排除。包含信号通常可同时从寄存器中获取。随后，或门的输出被送入一个宽输入与门；为减少路径延迟，该与门通常由多个较小的与门以树形结构连接实现。显然，对于某个固定的应用，若包含的文字数量较少，就可以将或门数量减少到$\mathbf{b}$，并使用仅有$\mathbf{b}$个输入的与门。这样可以降低硬件复杂度和功耗。对于一般情况，如果 Tsetlin 机需要能够针对不同应用进行编程，就需要有足够数量的可用文字。尽管如此，CSC-TM 仍能降低开关活动，从而节省功耗。

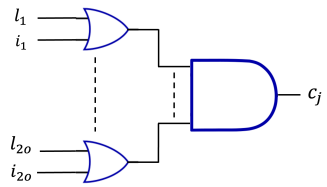


图3：子句逻辑的硬件实现 c.

需要注意的是，限制子句大小只会影响子句逻辑相关的能耗。后续的硬件处理（例如子句加权和求和）不受影响。不过，对于子句数量庞大的系统，子句逻辑会占用大量数字电路，而降低其开关活动可以显著节省能耗。

文字预算在系统层面的一项重要优势，是将模型从外部或片上内存加载到专用集成电路或现场可编程门阵列中的寄存器所需的时间。在此期间，系统的处理器和数据传输本身都会消耗能量。减少包含的文字数量后，采用合适的编码方案（例如游程编码 (RLE)），即可缩小模型大小，从而缩短加载时间 [Bakar 等， 2022a]。缩小模型大小对其他基于低功耗微控制器的嵌入式 TM 解决方案也有益。

6 结论与未来工作

本文提出了 CSC-TM——一种通过约束子句大小实现的新型 TM 机制。我们论证了，限制文字数量可以显著减少硬件中的开关活动，从而降低功耗。我们还分析并验证了 CSC-TM 的收敛性。实验结果表明，CSC-TM 能够达到

相同甚至更高的准确率，同时使用更短的子句，从而提升可解释性。未来的工作包括引入其他类型的约束，以支持更广泛的约束机器学习。

参考文献

[Abeyrathna等，2020年a] K. Darshana Abeyrathna、Ole- Christoffer Granmo、Rishad Shafik、Alex Yakovlev、Adrian Wheeldon、Jie Lei 和 Morten Goodwin。《一种用于任意确定性 Tsetlin 机器学习的新型多步有限状态自动机》。载于《计算机科学讲义：人工智能创新技术与应用第*40*届国际会议论文集（*SGAI-2020*）》。Springer International Publishing，2020年。[Abeyrathna 等，2020年b] K. Darshana Abeyrathna、Ole- Christoffer Granmo、Xuan Zhang、Lei Jiao 和 Morten Goodwin。《回归Tsetlin机——一种可解释非线性回归的新方法》。《英国皇家学会哲学汇刊 *A* 辑》，378，2020年。[Abeyrathna 等，2020年c] Kuruge Darshana Abeyrathna、Ole-Christoffer Granmo 和 Morten Goodwin。《通过整数加权子句扩展Tsetlin机器以增强可解释性》。*arXiv*预印本 *arXiv:2005.05131*，2020年。[Abeyrathna 等， 2021] K. Darshana Abeyrathna、Bimal Bhattarai、Morten Goodwin、Saeed Gorji、Ole-Christoffer Granmo、Lei Jiao、Rupsa Saha 和 Rohan K. Yadav。《支持近乎恒定时间扩展的大规模并行异步Tsetlin机器架构》。载于*ICML*，2021年。[Abouzeid 等， 2022] Ahmed Abouzeid、Ole-Christoffer Granmo、Morten Goodwin 和 Christian Webersik。《标签批评Tsetlin机：一种用于可解释聚类的新型自监督学习方案》。载于*Tsetlin*机国际研讨会（*ISTM*），第41–48页。IEEE，2022年。[Bakar 等，2022年a] A. Bakar、T. Rahman、A. Montanari、J. Lei、R. Shafik 和 F. Kawsar。《面向无电池传感器的逻辑智能》。载于移动计算系统与应用年度国际研讨会（*HotMobile*），第22–28页。计算机协会，2022年。[Bakar 等，2022年b] Abu Bakar、Tousif Rahman、Rishad Shafik、Fahim Kawsar 和 Alessandro Montanari。《利用软件加速Tsetlin机器实现无电池传感器的自适应智能》。载于*SenSys 2022*论文集。ACM，2022年。[Bhattarai等，2022年a] Bimal Bhattarai、Ole-Christoffer Granmo 和 Lei Jiao。《用于假新闻检测及可信度评分评估的可解释Tsetlin机框架》。载于*LREC*，2022年。[Bhattarai等，2022年b] Bimal Bhattarai、Ole-Christoffer Granmo 和 Lei Jiao。《使用Tsetlin机器进行新颖文本检测的词级人类可解释评分机制》。《应用智能》，2022年。[Bhattarai 等人， 2023] Bimal Bhattarai、Ole-Christoffer Granmo、Lei Jiao、Rohan Yadav 和 Jivitesh Sharma。《Tsetlin 机器嵌入：使用逻辑表达式表示词语》。arXiv 预印本 arXiv:2301.00709，2023 年。[Borgersen 等人， 2022] Karl Audun Kagnes Borgersen、Morten Goodwin 和 Jivitesh Sharma。《推荐系统背景下 Tsetlin 机器与深度神经网络的比较》。arXiv 预印本 arXiv:2212.10136，2022年。[Chang 等人， 2002] Eric Chang、Frank Seide、Helen M Meng、Zhuoran Chen、Yu Shi 和 Yuk-Chi Li。《移动设备上的语音查询信息检索系统》。IEEE 语音与音频处理汇刊，10(8):531–541，2002年。[Dally 等人， 2016] William J. Dally、Harting R. Curtis 和 Tor M. Aamodt。《使用 VHDL 的数字设计：系统方法》。剑桥大学出版社，2016年。[Debole 和 Sebastiani， 2005] Franca Debole 和 Fabrizio Sebastiani。《路透社 21578 子集相对难度分析》。美国信息科学与技术学会期刊，56(6):584–596，2005年。[Giri 等人， 2022] Charul Giri、Ole-Christoffer Granmo、Herke Van Hoof 和 Christian D. Blakely。《基于逻辑的人工智能：使用 Tsetlin 机器进行可解释的棋盘游戏赢家预测》。载于 IJCNN，第 1–9 页，2022年。[Glimsdal 和 Granmo， 2021] Sondre Glimsdal 和 Ole-Christoffer Granmo。《具有子句共享的合并式多输出 Tsetlin 机器》。arXiv 预印本，arXiv:2108.07594，2021年。[Glimsdal 等人， 2022] Sondre Glimsdal、Rupsa Saha、Bimal Bhattarai、Charul Giri、Jivitesh Sharma、Svein Anders Tunheim 和 Rohan Kumar Yadav。《聚焦负采样：提升 Tsetlin 机器的判别能力》。载于 2022年 Tsetlin 机器国际研讨会（ISTM），第 73–80 页，2022年。[Granmo 等人， 2019] Ole-Christoffer Granmo、Sondre Glimsdal、Lei Jiao、Morten Goodwin、Christian W. Omlin 和 Geir Thore Berge。《卷积 Tsetlin 机器》。arXiv 预印本 arXiv:1905.09688，2019年。[Granmo， 2018] Ole-Christoffer Granmo。《Tsetlin 机器——一种由博弈论驱动的多臂老虎机方法：基于命题逻辑实现最优模式识别》。arXiv 预印本 arXiv:1804.01508，2018年。[Greene 和 Cunningham， 2006] Derek Greene 和 P´adraig Cunningham。《核文档聚类中对角占优问题的实用解决方案》。载于 ICML，第 377–384 页。ACM 出版社，2006年。[Hendrickx 等人， 2009] Iris Hendrickx、Su Nam Kim、Zor-´nitsa Kozareva、Preslav Nakov、Diarmuid O S´eaghdha、Se- bastian Pad´o、Marco Pennacchiotti、Lorenza Romano 和 Stan Szpakowicz。《SemEval-2010 任务 8：名词短语对之间语义关系的多类别分类》。载于语义评测研讨会论文集：近期成果与未来方向，第 94–99 页。计算语言学协会，2009年。

[Jiao 等人， 2021] Lei Jiao、Xuan Zhang 和 Ole-Christoffer Granmo。《Tsetlin 机器在与运算符和或运算符下的收敛性》。*arXiv* 预印本 *https://arxiv.org/abs/2109.09488*，2021年。

[Jiao等，2022年8月 2022] Lei Jiao、Xuan Zhang、Ole- Christoffer Granmo 和 K Darshana Abeyrathna。《Tsetlin 机器异或运算符的收敛性》。*IEEE* 模式分析与机器智能汇刊，2022年8月。

[Lei等， 2021] Jie Lei、Tousif Rahman、Rishad Shafik、Adrian Wheeldon、Alex Yakovlev、Ole-Christoffer Granmo、Fahim Kawsar 和 Akhil Mathur。《使用 Tsetlin 机器进行低功耗音频关键词检测》。《低功耗电子与应用期刊》，11，2021年。

[Maas 等， 2011] Andrew Maas、RaymondE Daly、PeterT Pham、Dan Huang、Andrew Y Ng 和 Christopher Potts。《用于情感分析的词向量学习》。载于*ACL*，第142–150页，2011年。

[Saha等，2021] Rupsa Saha、Ole-Christoffer Granmo 和 Morten Goodwin。利用 Tsetlin 机器在自然语言处理应用中发现可解释规则。《专家系统》，第 e12873 页，2021年。

[Saha等，2022] R. Saha、O.-C. Granmo、V.I. Zadorozhny 和 M. Goodwin。一种用于自然语言理解的关系型 Tsetlin 机。《期刊 *o*》 f 《智能信息系统》，2022年。

[Seraj等，2022] Raihan Seraj、Jivitesh Sharma 和 Ole Christoffer Granmo。用于解决上下文老虎机问题的 Tsetlin 机器。发表于*NeurIPS*，2022年。

[Sharma等，2023] Jivitesh Sharma、Rohan Ku mar Yadav、Ole-Christoffer Granmo Granmo 和 Lei Jiao。Drop clause：提升 Tsetlin 机器的性能、鲁棒性和模式识别能力。发表于*AAAI*，2023 年。

[Tsetlin， 1961] Michael Lvovitch Tsetlin。《随机介质中有限自动机的行为》。《自动化与遥测》，22(10):1345–1355，1961年。

[Valiant， 1984] Leslie GValiant。《可学习性理论》。《ACM 通讯》，27(11):1134–1142，1984年。

[Wheeldon等， 2020] Adrian Wheeldon、Rishad Shafik、Tousif Rahman、Jie Lei、Alex Yakovlev 和 Ole-Christoffer Granmo。《基于学习自动机的物联网节能 AI 硬件设计》。《皇家学会哲学汇刊 A 辑》，2020年。

[Yadav 等，2021a] Rohan Yadav、Lei Jiao、Ole-Christoffer Granmo 和 Morten Goodwin。《利用预训练词语表示增强子句的语义可解释性》。载于第四届自然语言处理神经网络分析与解读研讨会，2021年。

[Yadav等，2021b] Rohan Yadav、Lei Jiao、Ole-Christoffer Granmo 和 Morten Goodwin。《用于基于方面的情感分析的人类水平可解释学习》。载于*AAAI*，2021年。

[Yadav等， 2022] Rohan Kumar Yadav、Lei Jiao、Ole Christoffer Granmo 和 Morten Goodwin。《使用带否定的与规则抵御虚假相关性的鲁棒可解释文本分类》。载于*IJCAI*，2022年。

Tsetlin 机器基础

分类

TM 将一个由布尔特征组成的向量 $X = [x_1, \ldots, x_o]$ 作为输入，并将其分类为两个类别之一： $y = 0$ 或$y = 1$。这些特征会转换为一组文字，其中包括特征本身及其取反对应项，如$L = \{x_1, \ldots, x_o, \neg x_1, \ldots, \neg x_o\}$所示。

如果类别数量为 q ，子模式数量为 n ，则 TM 模式使用 $q \times n$ 个合取子句来构造， $1 \leq j \leq n$ 表示为：

$$C_j(X) = \left(\bigwedge_{l_k \in L_j} l_k\right) \bigwedge \left(\bigwedge_{l_k \in \bar{L}_j} l_k\right), \tag{5}$$

其中， L_j 是文字集合 L 的子集，包含原始形式的文字；而 L_j 是 L 的子集，包含取反形式的文字。例如，子句 $C_j(X) = x_1 \wedge x_2 = x_1 x_2$ 由文字 $L_j = \{x_1, x_2\}$、$L_j = \{\neg x_1, \neg x_2\}$ 组成；如果 $x_1 = x_2 = 1$，则输出 1。

所使用的子句数量是由用户设定的参数n。一半的子句被赋予正极性，其索引为奇数。另一半被赋予负极性，其索引为奇数。子句输出通过求和，并使用单位阶跃函数 $u(v) = 1$**若** $v \geq 0$ **否则** 0 进行阈值处理，从而得出分类决策：

$$\hat{y} = u\left(\sum_{j=1}^{n/2} C_j^+(X) - \sum_{j=1}^{n/2} C_j^-(X)\right). \tag{6}$$

也就是说，分类基于多数投票进行，正子句为 $y = 1$ 投票，负子句则为 $y = 0$投票。

学习

一个子句 $C_j(X)$ 由一组 Tsetlin 自动机组成， [Tsetlin, 1961], 每个自动机都会决定 包含 或 排除 子句中的某个特定文字 l_k 。Tsetlin 自动机根据所接收的反馈作出决策，反馈形式包括奖励、不动作和惩罚。TM 学习涉及两种反馈类型：I型反馈和 II 型反馈，如表 6 和表 7所示。

I型反馈 以随机方式提供给正极性的子句（当 $y = 1$ 时）以及负极性的子句（当 $y = 0$时）。每个子句则根据以下因素强化其 Tsetlin 自动机：(1) 其输出 $C_j(X)$；(2) Tsetlin 自动机的动作——包含 或 排除；以及 (3) 分配给该 Tsetlin 自动机的文字 l_k 的值。I 型反馈受两条规则支配：

- 包含 会受到奖励，而排除 会以概率 $\frac{s-1}{s}$ **如果** $C_j(X) = 1$ **并且** $l_k = 1$ 受到惩罚。这种强化很强（以高概率触发），会使子句记住并细化其在 X 中识别的模式。[7]

[7]请注意，提升真正例时，概率 $\frac{s-1}{s}$ 会替换为 1。

• 包含 会受到惩罚，而排除 会以概率 $\frac{1}{s}$ **如果** $C_j(X) = 0$ **或** $l_k = 0$受到奖励。这种强化较弱（以低概率触发），会使不频繁的模式变得粗略，从而使其变得频繁。

上文中的参数 s控制模式频率。

II 型反馈 会以随机方式反馈给具有正极性的子句（当 $y = 0$ 时），以及具有负极性的子句（当 $y = 1$ 时）。当满足 $C_j(X) = 1$ **且** $l_k = 0$ 时，该反馈会以概率 **1惩罚** 排除。因此，此反馈会生成用于区分 $y = 0$ 和 $y = 1$ 的文字。

子句的值 $C_j^i(\mathbf{X})$		1		0	
文字的值 $x_k/\neg x_k$		1	0	1	0
Tsetlin 自动机：**包含**	P(奖励)	$\frac{s-1}{s}$	NA	0	0
	P(不动作)	$\frac{1}{s}$	NA	$\frac{s-1}{s}$	$\frac{s-1}{s}$
文字	P(惩罚)	0	NA	$\frac{1}{s}$	$\frac{1}{s}$
Tsetlin 自动机：**排除**	P(奖励)	0	$\frac{1}{s}$	$\frac{1}{s}$	$\frac{1}{s}$
	P(不动作)	$\frac{1}{s}$	$\frac{s-1}{s}$	$\frac{s-1}{s}$	$\frac{s-1}{s}$
文字	P(惩罚)	$\frac{s-1}{s}$	0	0	0

表 6：原始TM 的 I型反馈——收到标签为 $y = 1$的样本后，为单个 Tsetlin 自动机提供反馈，以决定是否将给定文字 $x_k/\neg x_k$ 包含到 C_j^i中。不适用表示不适用。

子句的值 $C_j^i(\mathbf{X})$		1		0	
文字的值 $x_k/\neg x_k$		1	0	1	0
Tsetlin 自动机：**包含**	P(奖励)	0	NA	0	0
	P(不动作)	1.0	NA	1.0	1.0
文字	P(惩罚)	0	NA	0	0
Tsetlin 自动机：**排除**	P(奖励)	0	0	0	0
	P(不动作)	1.0	0	1.0	1.0
文字	P(惩罚)	0	1.0	0	0

表 7：II 型反馈 — 收到标签为 $y = 0$ 的样本后，对单个 Tsetlin 自动机进行反馈，以决定是否将给定文字 $x_k/\neg x_k$ 包含到 C_j^i中。不适用表示不适用。

让我们考虑一个异或门样本（$x_1 = 0, x_2 = 1$）=1，如图 4 所示，以直观展示学习过程。学习异或模式需要 n 个子句，这里我们考虑每个类别 $n = 4$ 个。在 4 个子句中，子句C_1 和 C_3 对其存在投票，$y = 1$ 和 C_0 和 C_2 则投票反对。为简化起见，我们只考虑 C_1 和C_3 如何针对给定的异或门样本学习该模式。步骤1中，子句尚未学会给定样本的模式，因此导致类别预测错误，进而触发对相应文字的 I 型反馈。根据表 6，对于文字 x_1，如果子句得分为 0 且文字为 0，则在被包含时分别以 $\frac{s-1}{s}$ 和 $\frac{1}{s}$ 的概率收到不动作或惩罚。经过多次惩罚后，它会改变状态，转为排除动作，并从子句C_1中移除。另一方面，文字 $\neg x_1$ 因被排除而受到惩罚，最终跳转到包含区段，如 C_1 所示。类似地，当文字 $\neg x_2 = 0$ 和 C_1 =为 0 时，它在被包含时会收到不动作或惩罚，
$\frac{s-1}{s}$ 和 $\frac{1}{s}$ 的概率分别为。在多次惩罚后，

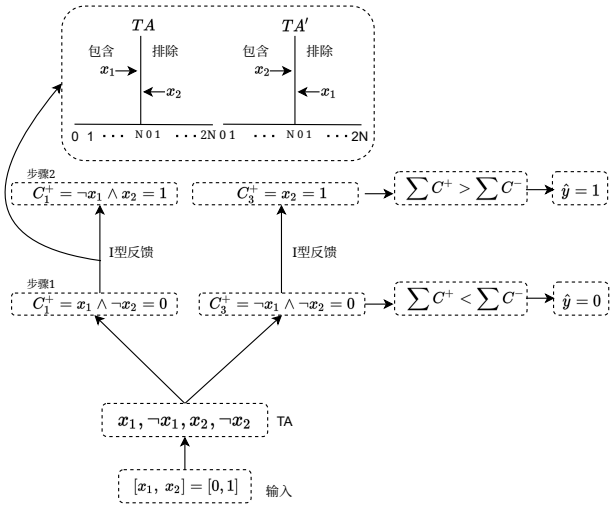


图 4：Tsetlin 机器对异或门样本的学习。

x_1	x_2	输出
0	0	0
1	1	0
0	1	1

表8：“异或”情形下的一个子模式。

$\neg x_2$ 被排除，而 x_2 如步骤2所示变为包含状态。这确实达到了预期的模式，从而使子句变为 $C_1 = 1$和 $C_3 = 1$，最终得到$\hat{y} = 1$。

资源分配可确保子句分散到各个频繁模式中，而不是遗漏某些模式，并过度集中于其他模式。也就是说，对于任意输入 X，随着子句输出和逐渐增加，强化该子句的概率会逐渐降至零。

$$v = \sum_{j=1}^{n/2} C_j^+(X) - \sum_{j=1}^{n/2} C_j^-(X) \tag{7}$$

接近用户设定的目标值 T用于 $y = 1$(以及 $-T$用于$y = 0$）。如果某个子句未得到强化，则不会向其 Tsetlin 自动机提供反馈，因此它们会保持不变。极端情况下，当投票和 v 等于或超过目标值 T（TM 已成功识别输入 X）时，没有子句会得到强化。此时，它们可以自由学习新的模式，自然而然地平衡模式表示资源 [Granmo, 2018]。

B 在给定新约束下 XOR 子模式的详细状态转移

本节讨论仅给定一个文字时异或运算符的收敛情况，即（$\|C_j^i(\mathbf{X})\| = 1$）。

这里我们研究带有长度约束的 Tsetlin 机中 Tsetlin 自动机的详细状态转移，输入为表8中所示的异或子模式。与 [Jiao 等人，Aug2022], 的分析相比，新约束带来的变化以红色标出。

不失一般性，我们研究子句 C_3，它具有 Tsetlin 自动机$_1^3$，其动作是“包含” x_1 或“排除”它；另一个 Tsetlin 自动机$_2^3$ 的动作是

“包含” $\neg x_1$ 或“排除”它；一个 Tsetlin 自动机$_3^3$ 的动作是“包含” x_2 或“排除”它，另一个 Tsetlin 自动机$_4^3$ 的动作是“包含”$\neg x_2$ 或“排除”它。为了分析这四个 Tsetlin 自动机的收敛情况，我们进行准平稳分析：固定其中三个自动机的行为，然后研究剩下一个自动机的状态转移。具体而言，分析安排如下：

1. 我们分别将 TA_1^3 和 TA_2^3 冻结为“排除”和“包含”。在这种情况下，第一位变为 $\neg x_1$。TA_3^3 和 TA_4^3有四种子情况：

   (a) 我们研究 TA_3^3 当前动作是“包含”时的状态转移，给定表8 所示的不同训练样本，以及 TA_4^3 的不同动作（即 TA_4^3 的动作被冻结为“包含”或“排除”时）。

   (b) 我们研究TA_3^3 当前动作为“排除”时的状态转移，给定表 8 所示的不同训练样本，以及TA_4^3 的不同动作（即 TA_4^3 的动作被冻结为“包含”或“排除”时）。

   (c) 我们研究 TA_4^3 当前动作为“包含”时的状态转移，给定表 8 所示的不同训练样本，以及TA_3^3 的不同动作（即 TA_3^3 的动作被冻结为“包含”或“排除”时）。

   (d) 我们研究Tsetlin 自动机$_4^3$ 在当前动作是“排除”时，给定表中所示的不同训练样本8 以及Tsetlin 自动机$_3^3$ 的不同动作（即Tsetlin 自动机$_3^3$ 的动作冻结为“包含”或“排除”时）的状态转移。

2. 我们分别将 TA_1^3 和 TA_2^3 冻结在“包含”和“排除”状态。在这种情况下，第一个位变为 x_1。TA_3^3 和 TA_4^3 的子情况与前一种情况中的子情况相同。

3. 我们将 TA_1^3 和 TA_2^3 冻结在“排除”和“排除”状态。在这种情况下，第一个位被排除，不会影响最终输出。TA_3^3 和 TA_4^3 的子情况与前一种情况中的子情况相同。

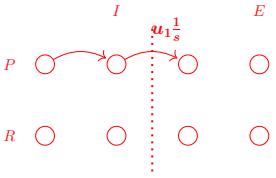
4. 我们将 TA_1^3 和 TA_2^3 冻结在“包含”和“包含”状态。在这种情况下，由于子句包含矛盾 $x_1 \wedge \neg x_1$，因此结果始终为 $C_3 = 0$。TA_3^3 和 TA_4^3 的子情况与前一种情况中的子情况相同。

在下面的分析中，我们将逐一研究这四种情况。

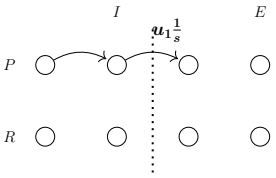
情况 1

我们分析第一个子情形，即子情形 1 (a)。在此情形中，第一个位始终为 $\neg x_1$。这里我们研究当前动作是“包含”的 Tsetlin 自动机 $_3^3$ 的状态转移。根据不同的训练样本和 Tsetlin 自动机 $_4^3$, 的动作，我们得到以下可能的状态转移。下文中，“I”和“E”分别表示“包含”和“排除”。为简洁起见，我们省略了未发生状态转移的实例。

条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机 $_4^3$= 排除。因此，对于文字 $x_2 = 1$, $C_3 = \neg x_1 \wedge x_2 \wedge 0 = 0$, 我们得到 I 型反馈。

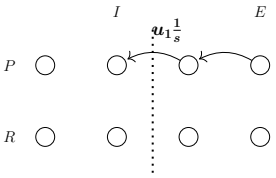


条件：$x_1 = 0, x_2 = 1$, $y = 1$, TA_4^3=I。因此，对于文字 $x_2 = 1$, $C_3 = 0$, 我们有 I 型反馈。

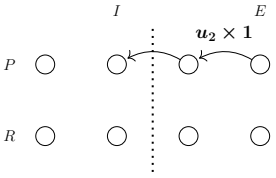


现在考虑子情形 1(b)。文字 $\neg x_1$ 仍然被包含，我们来研究 TA_3^3 在其当前动作为“排除”时的状态转移。可能的转移如下所列。

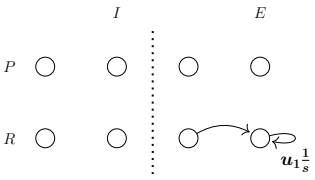
条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, TA_4^3=E。因此，Type I, $x_2 = 1$, $C_3 = \neg x_1 = 1$.



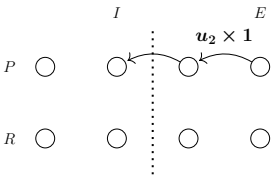
条件：$x_1 = 0$, $x_2 = 0$, $y = 0$, Tsetlin 自动机 $_4^3$=E。因此，Type II, $x_2 = 0$, $C_3 = \neg x_1 = 1$。



条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机 $_3^3$=I。因此，Type I, $x_2 = 1$, $C_3 = \neg x_1 \wedge \neg x_2 = 0$.

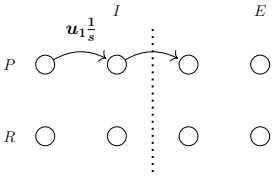


条件：$x_1 = 0$, $x_2 = 0$, $y = 0$, Tsetlin 自动机 $_3^3$=I。因此，Type II, $x_2 = 0$, $C_3 = \neg x_1 \wedge \neg x_2 = 1$。

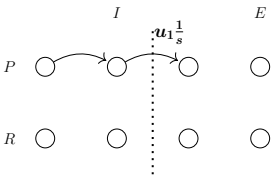


现在我们来看情况 1 中的第三个子情况，即子情况 1 (c)。文字 $\neg x_1$ 仍然被包含，我们来研究 Tsetlin 自动机$_4^3$ 当前动作为“包含”时的状态转移。请注意，我们现在研究的是 Tsetlin 自动机$_4^3$ 对应于 $\neg x_2$而非 x_2的情况。因此，表 6 和 7 中的文字变为 $\neg x_2$。

Condition: $x_1 = 0$, $x_2 = 1$, $y = 1$, TA_3^3=E. Therefore, Type I, $\neg x_2 = 0$, $C_3 = \neg x_1 \wedge \neg x_2 = 0$.

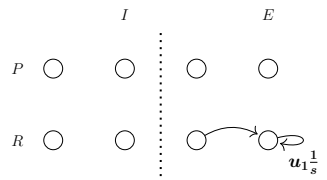


Condition: $x_1 = 0$, $x_2 = 1$, $y = 1$, TA_3^3=I. Therefore, Type I, $\neg x_2 = 0$, $C_3 = 0$.

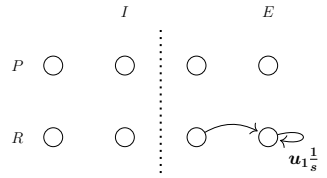


对于子情况 1 (d)，我们研究 Tsetlin 自动机$_4^3$当前动作为“排除”时的状态转移。

条件：$x_1 = 0, x_2 = 1, y = 1,$ Tsetlin 自动机$_3^3$=E。
因此，Type I，$\neg x_2 = 0, C_3 = \neg x_1 = 1$.



条件：$x_1 = 0, x_2 = 1, y = 1,$ Tsetlin 自动机$_3^3$=I。
因此，Type I，$\neg x_2 = 0$，$C_3 = \neg x_1 \wedge x_2 \wedge 0 = 1$。



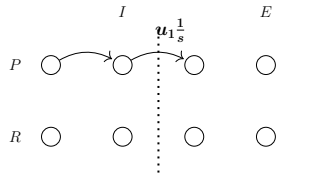
到目前为止，我们已经讨论完情况 1 中的所有子情况。

情况 2

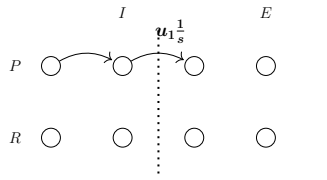
情况 2 研究 TA_3^3 和 TA_4^3 的行为，其中 TA_1^3 和 TA_2^3 分别选择“包含”和“排除”。在这种情况下，第一个位始终为 x_1 的形式。这里同样有四个子情况，我们将在下文逐一详细说明。

我们首先研究执行“包含”动作的 TA_3^3，并给出以下状态转移。

条件：$x_1 = 0, x_2 = 1, y = 1,$ Tsetlin 自动机$_4^3$=E。
因此，Type I，$x_2 = 1, C_3 = 0$。

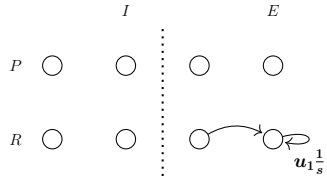


条件：$x_1 = 0, x_2 = 1, y = 1,$ Tsetlin 自动机$_4^3$=I。
因此，Type I，$x_2 = 1, C_3 = 0$。

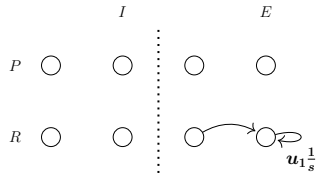


然后，我们研究执行“排除”动作的 TA_3^3，其状态转移如下所示。

条件：$x_1 = 0, x_2 = 1, y = 1,$ TA_4^3=E。
因此，Type I，$x_2 = 1, C_3 = 0$。

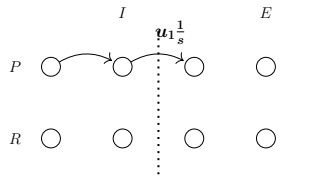


条件：$x_1 = 0, x_2 = 1, y = 1,$ Tsetlin 自动机$_4^3$=I。
因此，Type I，$x_2 = 1, C_3 = 0$。

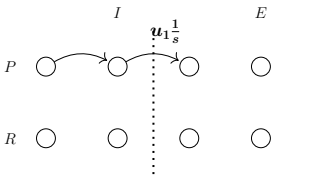


现在我们研究具有“包含”动作的 Tsetlin 自动机$_4^3$，其状态转移如下所示。

条件：$x_1 = 0, x_2 = 1, y = 1,$ Tsetlin 自动机$_3^3$=E。因此，I 型，$\neg x_2 = 0, C_3 = x_1 \wedge \neg x_2 = 0$。

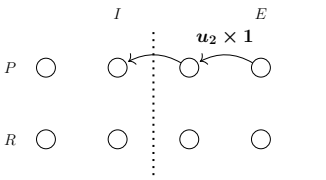


条件：$x_1 = 0, x_2 = 1, y = 1,$ TA_3^3=I。因此，I 型，$\neg x_2 = 0, C_3 = 0$。

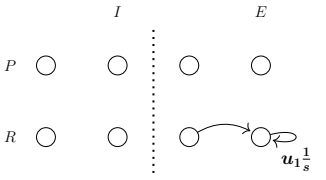


最后，我们研究动作“排除”的 TA_4^3，它会产生以下状态转移。

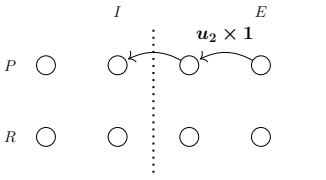
条件：$x_1 = 1, x_2 = 1, y = 0,$ TA_3^3=E。
因此，Type II，$\neg x_2 = 0, C_3 = x_1 = 1$。



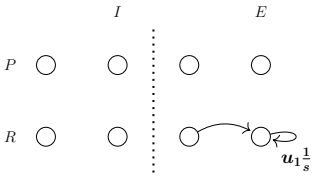
条件：$x_1 = 0, x_2 = 1, y = 1,$ Tsetlin 自动机$_3^3$=E。
因此，Type I，$\neg x_2 = 0, C_3 = 0$。



条件：$x_1 = 1, x_2 = 1, y = 0,$ Tsetlin 自动机$_3^3$=I。
因此，Type II，$\neg x_2 = 0, C_3 = x_1 \wedge x_2 = 1$。



条件：$x_1 = 0, x_2 = 1, y = 1,$ Tsetlin 自动机$_3^3$=I。
因此，Type I，$\neg x_2 = 0, C_3 = x_1 \wedge x_2 = 0$。

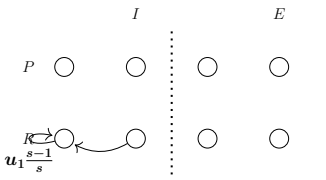


情况 3

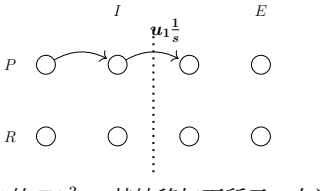
现在我们进入情况 3，其中 TA_1^3 和 TA_2^3 都选择“排除”。我们研究 TA_3^3 和 TA_4^3 在不同子情况下的行为。在这种情况下，第一个位 x_1 对输出不起任何作用。

我们先考察动作是“包含”的 TA_3^3，其状态转移如下。

条件：$x_1 = 0, x_2 = 1, y = 1,$ TA_4^3=E。因此，I 型，$x_2 = 1, C_3 = x_2 = 1$。

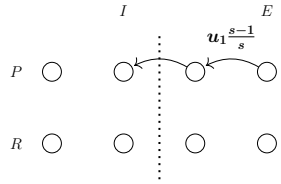


条件：$x_1 = 0, x_2 = 1, y = 1,$ TA_4^3=I。因此，I 型，$x_2 = 1, C_3 = 0$。

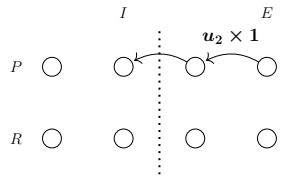


然后我们研究动作是“排除”的 TA_3^3，其转移如下所示。在这种情况下，如果 TA_4^3 也被排除，那么由于与 C_3 相关联的所有 TA 都选择了“排除”这一动作，因此 C_3 是“空的”。为使训练继续进行，我们根据 TM 的训练规则，在这种情况下赋予 $C_3 = 1$。

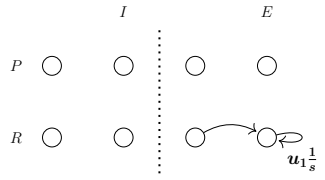
条件： $x_1 = 0$、$x_2 = 1$、$y = 1$，TA_4^3=E。因此，I 型、$x_2 = 1$、$C_3 = 1$。



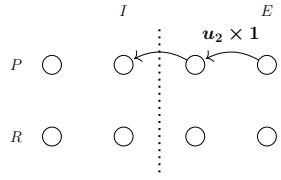
条件： $x_1 = 0$，$x_2 = 0$，$y = 0$，Tsetlin 自动机$_4^3$=E。因此，II型,$x_2 = 0, C_3 = 1$。



条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_4^3$=I。因此，Type I，$x_2 = 1$，$C_3 = \neg x_2 = 0$。

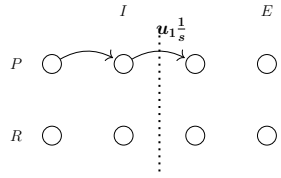


条件： $x_1 = 0$，$x_2 = 0$，$y = 0$，Tsetlin 自动机$_4^3$=I。因此，Type II，$x_2 = 0$，$C_3 = \neg x_2 = 1$。

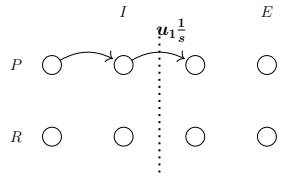


第三，我们研究执行“包含”动作的 Tsetlin 自动机$_4^3$，涵盖下方所示的状态转移。

条件： $x_1 = 0$，$x_2 = 1$，$y = 1$ Tsetlin 自动机$_3^3$=E.因此，Type I，$\neg x_2 = 0, C_3 = 0$。

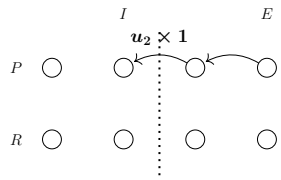


条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_3^3$=I.因此，Type I，$\neg x_2 = 0, C_3 = 0$。

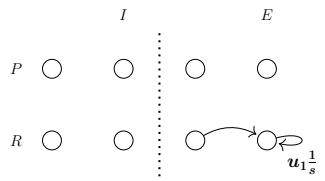


最后，我们研究动作是“排除”的 Tsetlin 自动机$_4^3$，其状态转移如下所示。同样，在这种情况下，当 Tsetlin 自动机$_3^3$ 也被排除时，C_3 会再次变为“空”，因为与其关联的所有 Tsetlin 自动机都选择了动作“排除”。按照 Tsetlin 机的训练规则，我们赋予 $C_3 = 1$。

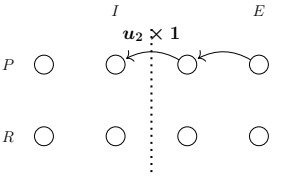
条件： $x_1 = 1$，$x_2 = 1$，$y = 0$，Tsetlin 自动机$_3^3$=E.因此，Type II，$\neg x_2 = 0, C_3 = 1$.



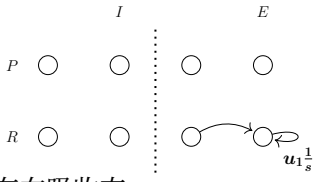
条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_3^3$=E.因此，Type I，$\neg x_2 = 0, C_3 = 1$。



条件： $x_1 = 1$，$x_2 = 1$，$y = 0$，Tsetlin 自动机$_3^3$=I。因此，Type II，$\neg x_2 = 0, C_3 = 1$。



条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_3^3$=I.因此，Type I，$\neg x_2 = 0, C_3 = 1$。

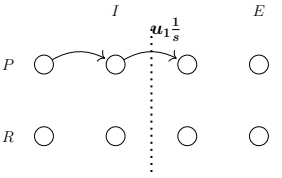


显然，在情况 3 中，不存在吸收态。

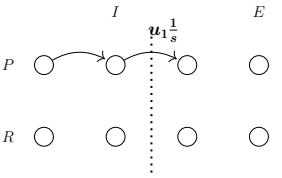
情况 4

现在，我们研究情况 4，其中 $\neg x_1$ 和 x_1 都选择“包含”。因此，在这种情况下，我们总是有 $C_3 = 0$。我们首先研究动作是“包含”的 Tsetlin 自动机$_3^3$，其状态转移如下所示。

条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_3^3$=E.因此，Type I，$x_2 = 1, C_3 = 0$.

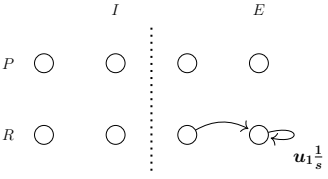


条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_4^3$=I.因此，Type I，$x_2 = 1, C_3 = 0$。

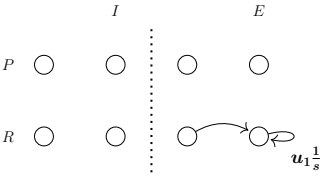


其次，我们研究动作“排除”的 TA_3^3。

条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，TA_4^3=E .因此，Type I，$x_2 = 1, C_3 = 0$。

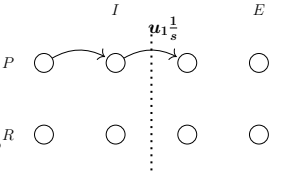


条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_4^3$=I.因此，Type I，$x_2 = 1, C_3 = 0$.

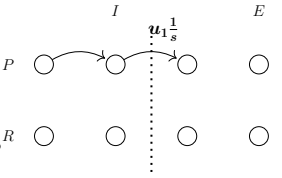


现在，我们研究执行“包含”动作的 Tsetlin 自动机$_4^3$。

条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_3^3$=E。因此，I 型,$\neg x_2 = 0, C_3 = 0$。

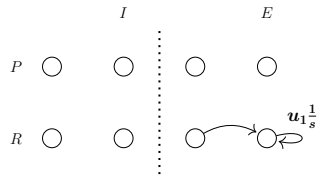


条件： $x_1 = 0$，$x_2 = 1$，$y = 1$，Tsetlin 自动机$_3^3$=I。因此，I 型,$\neg x_2 = 0, C_3 = 0$。

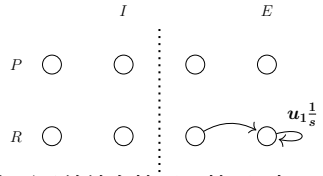


最后，我们研究动作“排除”的 Tsetlin 自动机$_4^3$。

条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机^3=E。
因此，Type I, $\neg x_2 = 0, C_3 = 0$。



条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$^3_3$=I。
因此，Type I, $\neg x_2 = 0, C_3 = 0$。



根据以上分析，我们现在可以总结在情况1–情况4中，Tsetlin 自动机$^3_1$ 和 Tsetlin 自动机$^3_2$ 处于不同配置时，Tsetlin 自动机$^3_3$ 和 Tsetlin 自动机$^3_4$, 的状态转移（也就是说，x_1 和 $\neg x_1$的四种不同组合）。下方的箭头表示转移方向。

场景1： 研究 Tsetlin 自动机$^3_3$ = I 和 Tsetlin 自动机$^3_4$ = I。

情况 1： 我们可以看到，$TA^3_3 \to E$ $TA^3_4 \to E$

情况 2： 我们可以看到 Tsetlin 自动机$^3_3$ →E Tsetlin 自动机$^3_4$ →E

情况 3： 我们可以看到
$TA^3_3 \to E$
$TA^3_4 \to E$

情况 4： 我们可以看到
$TA^3_3 \to E$
$TA^3_4 \to E$

根据上述事实可以确认，无论 Tsetlin 自动机$^3_1$ 和 Tsetlin 自动机$^3_2$, 处于什么状态，只要 Tsetlin 自动机$^3_3$=I 且 Tsetlin 自动机$^3_4$=I，它们（Tsetlin 自动机$^3_3$ 和 Tsetlin 自动机$^3_4$）就会远离当前状态，朝状态空间的另一半移动（即朝“排除”方向移动）。因此，Tsetlin 自动机$^3_3$=I 且 Tsetlin 自动机$^3_4$=I 的状态不是吸收态。

场景2： 考察 Tsetlin 自动机$^3_3$ = I 和 Tsetlin 自动机$^3_4$= E。

情况 1： 我们可以看到 $TA^3_3 \to E$, $TA^3_4 \to E$

情况 2： 我们可以看到 Tsetlin 自动机$^3_3$ →ETsetlin 自动机$^3_4$ →I, E

情况 3： 我们可以看到
$TA^3_3 \to I$
$TA^3_4 \to I, E$

情况 4： 我们可以看到
$TA^3_3 \to E$
$TA^3_4 \to E$

在此场景中，Tsetlin 自动机$^3_3$ 的起始点为“包含”，而 Tsetlin 自动机$^3_4$ 的起始点为“排除”。显然，Tsetlin 自动机$^3_3$ 和 Tsetlin 自动机$^3_4$ 的动作“包含”和“排除”都不是吸收态，因为所有情况都无法使 Tsetlin 自动机$^3_3$ 和 Tsetlin 自动机$^3_4$ 只朝“包含”和“排除”方向移动。

情景 3： 研究 Tsetlin 自动机$^3_3$ = E 和 Tsetlin 自动机$^3_4$ = I。

情况 1： 我们可以看到，$TA^3_3 \to I, E$ $TA^3_4 \to E$

情况 2： 我们可以看到 Tsetlin 自动机$^3_3$ →E Tsetlin 自动机$^3_4$ →E

情况 3： 我们可以看到
$TA^3_3 \to I, E$
$TA^3_4 \to E$

情况 4： 我们可以看到
$TA^3_3 \to E$
$TA^3_4 \to E$

根据情景 3 中 TA^3_3 和 TA_4 的转移，[3] 我们可以得出结论：具有 TA^3_3 =E 和 TA^3_4 =I 的状态不是吸收态。

Scenario 4: Study TA^3_3 = E and TA^3_4 = E.

情况 1： 我们可以看到
$TA^3_3 \to I$
$TA^3_4 \to E$

情况 2： 我们可以看到 Tsetlin 自动机$^3_3$→ E Tsetlin 自动机 $^3_4$ →I, E

情况 3： 我们可以看到
$TA^3_3 \to I$
$TA^3_4 \to I, E$

情况 4： 我们可以看到
$TA^3_3 \to E$
$TA^3_4 \to E$

根据场景4中 Tsetlin 自动机$_3$ 和 Tsetlin 自动机$_4$ 的状态转移，我们可以得出结论：在情况 4 中，当 Tsetlin 自动机$^3_1$ 和 Tsetlin 自动机$^3_2$ 的动作均为包含时，状态 Tsetlin 自动机$^3_3$ = E 和 Tsetlin 自动机$^3_4$ = E 也处于吸收态。

根据上述分析，我们可以得出结论：当我们将 Tsetlin 自动机$^3_1$ 和 Tsetlin 自动机$^3_2$ 固定为某些动作时，不存在吸收情况。尽管当 Tsetlin 自动机$^3_3$ 和 Tsetlin 自动机$^3_4$ 的动作均为“排除”时，情况看似是吸收的，但鉴于 Tsetlin 自动机$^3_1$ 的动作是“包含”且 Tsetlin 自动机$^3_2$ 的动作是“包含”，由于该条件实际上处于暂态，因此这并非真正的吸收态。

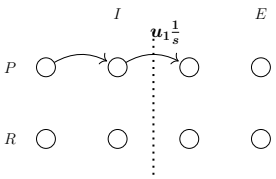
到目前为止，我们研究了在 TA^3_1 和 TA^3_2 的转移被冻结时，TA^3_3 和 TA^3_4的行为。接下来，我们冻结 TA^3_3 和 TA^3_4 的动作，并研究 TA^3_1 和 TA^3_2 的转移。

情况 1

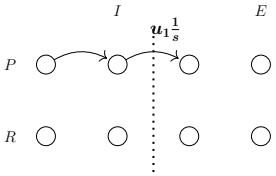
此处，TA^3_3 被冻结为“排除”，而 TA^3_4 为“包含”。在这种情况下，TA^3_3 和 TA^3_4 的输出为 $\neg x_2$。

我们首先研究动作是“包含”的TA^3_1。

条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, TA$^3_2$=E。因此，I 型，$x_1 = 0$, $C_3 = x_1 \wedge \neg x_2 = 0$。

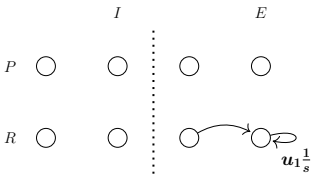


条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, TA^3_2=I。因此，I 型，$x_1 = 0, C_3 = 0$。

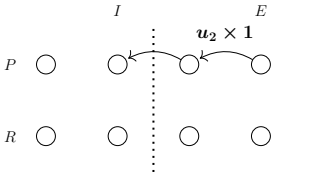


现在我们研究Tsetlin 自动机$^3_1$ **执行“排除”动作的情况。**

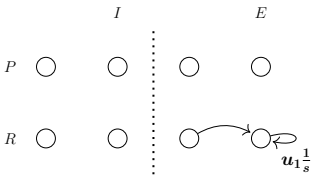
条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$^3_2$=E。因此，I 型，$x_1 = 0, C_3 = \neg x_2 = 0$。



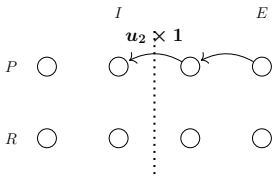
条件：$x_1 = 0$, $x_2 = 0$, $y = 0$, Tsetlin 自动机$^3_2$=E。因此，II型，$x_1 = 0, C_3 = \neg x_2 = 1$。



条件：$x_1 = 0$、$x_2 = 1$、$y = 1$、Tsetlin 自动机$^3_2$=I。因此，I 型，$x_1 = 0$、$C_3 = \neg x_1 \wedge \neg x_2 = 0$。

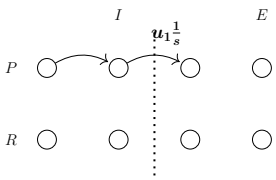


条件： $x_1 = 0$、$x_2 = 0$、$y = 0$、Tsetlin 自动机$_2^3$=I。因此，II型，$x_1 = 0$、$C_3 = \neg x_1 \wedge \neg x_2 = 1$。

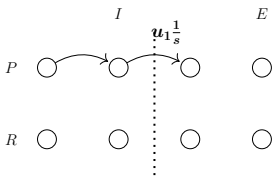


我们第三步研究Tsetlin 自动机$_2^3$ 的动作“包含”。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=E。因此，I 型，$\neg x_1 = 1$, $C_3 = \neg x_1 \wedge \neg x_2 = 0$

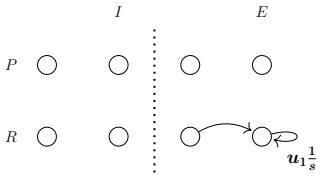


条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=I。因此，I 型，$\neg x_1 = 1$, $C_3 = 0$

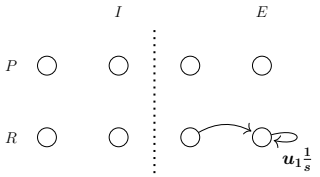


我们最后研究Tsetlin 自动机$_2^3$ 的动作“排除”。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=E。因此，I 型，$\neg x_1 = 1$, $C_3 = \neg x_2 = 0$



条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=I。因此，I 型，$\neg x_1 = 1$, $C_3 = x_1 \wedge \neg x_2 = 0$。

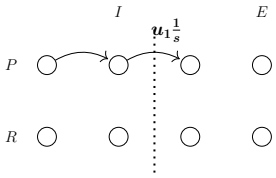


情况 2

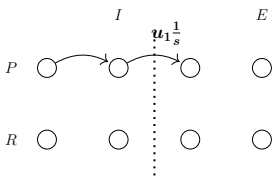
此时，TA$_3^3$ 被冻结为“包含”，而 TA$_4^3$ 为“排除”。在这种情况下，TA$_3^3$ 和 TA$_4^3$ 的输出给出 x_2。

现在我们研究Tsetlin 自动机$_1^3$，其动作是“包含”。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, TA$_2^3$=E。因此，I 型，$x_1 = 0$, $C_3 = x_1 \wedge x_2 = 0$。

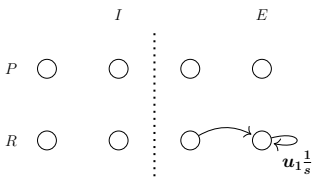


条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, TA$_2^3$=I。因此，I 型，$x_1 = 0$, $C_3 = \neg x_1 \wedge x_1 \wedge x_2 = 0$。

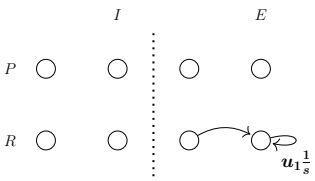


现在我们研究Tsetlin 自动机$_1^3$，其动作是“排除”。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=E。因此，I 型，$x_1 = 0$, $C_3 = x_1 = 1$。

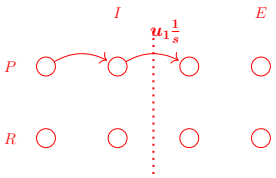


条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=I。因此，I 型，$x_1 = 0$, $C_3 = \neg x_1 \wedge x_2 \wedge 0 = 0$。

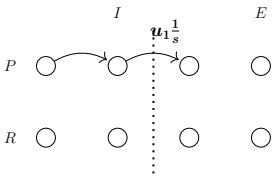


现在我们研究Tsetlin 自动机$_2^3$ 执行“包含”动作时的情况。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=E。因此，I 型，$\neg x_1 = 1$, $C_3 = \neg x_1 \wedge x_2 \wedge 0 = 0$。

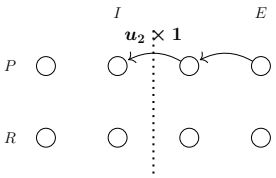


条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=I。因此，I 型，$\neg x_1 = 1$, $C_3 = 0$。

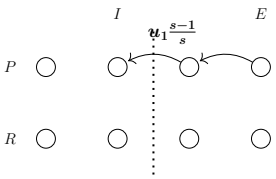


现在我们研究Tsetlin 自动机$_2^3$ 执行“排除”动作时的情况。

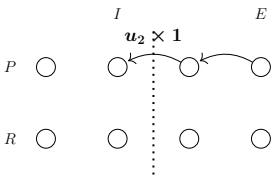
条件： $x_1 = 1$, $x_2 = 1$, $y = 0$, Tsetlin 自动机$_1^3$=E。因此，II型，$\neg x_1 = 0$, $C_3 = x_2 = 1$。



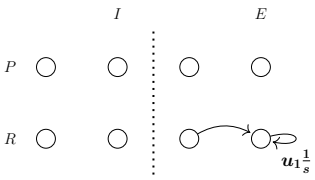
条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=E 因此，I 型，$\neg x_1 = 1$, $C_3 = x_2 = 1$。



条件： $x_1 = 1$, $x_2 = 1$, $y = 0$, TA$_1^3$=I。因此，II型，$\neg x_1 = 0$, $C_3 = x_1 \wedge x_2 = 1$。



条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, TA$_1^3$=I。因此，I 型，$\neg x_1 = 1$, $C_3 = x_1 \wedge x_2 = 0$。

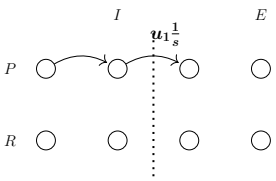


情况 3

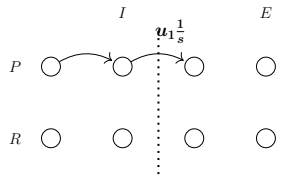
这里，TA$_3^3$ 被冻结为“排除”，TA$_4^3$ 也被冻结为“排除”。在这种情况下，第二个位 x_2 对输出不起任何作用。

现在我们研究Tsetlin 自动机$_1^3$ 执行“包含”动作的情况。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=E。因此，I 型，$x_1 = 0$, $C_3 = x_1 = 0$。

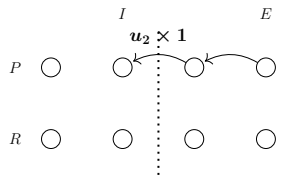


条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=I。因此，I 型，$x_1 = 0$, $C_3 = x_1 \wedge \neg x_1 = 0$。

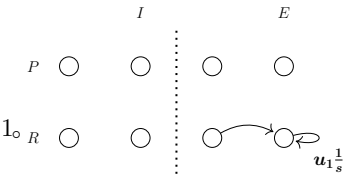


现在我们研究动作是“排除”的 Tsetlin 自动机$_1^3$。

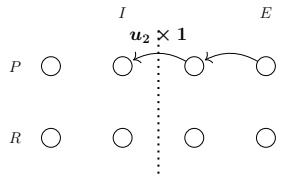
条件： $x_1 = 0$, $x_2 = 0$, $y = 0$, Tsetlin 自动机$_2^3$=E。因此，II型，$x_1 = 0$,$C_3 = 1$。



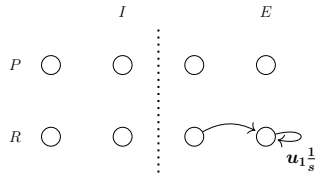
条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=E。因此，I 型，$x_1 = 0$,$C_3 = 1$。



条件： $x_1 = 0$, $x_2 = 0$, $y = 0$, Tsetlin 自动机$_2^3$=I。因此，II型，$x_1 = 0$, $C_3 = \neg x_1 = 1$。

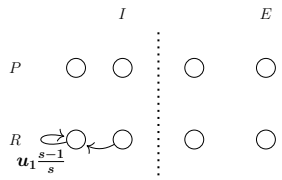


条件； $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=I。因此，I 型，$x_1 = 0$, $C_3 = \neg x_1 = 1$。

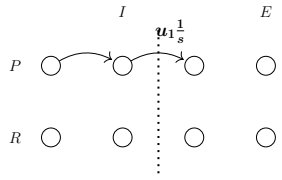


现在我们研究Tsetlin 自动机$_2^3$**，其动作是“包含”**。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=E。因此，I 型，$\neg x_1 = 1$, $C_3 = \neg x_1 = 1$。

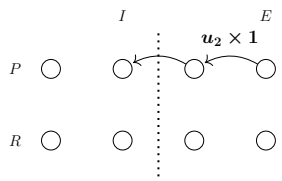


条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=I。因此，I 型，$\neg x_1 = 1$, $C_3 = 0$。

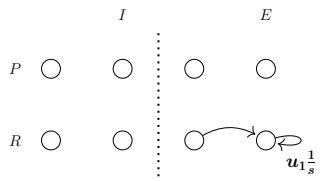


现在我们研究Tsetlin 自动机$_2^3$ **的“排除”动作**。

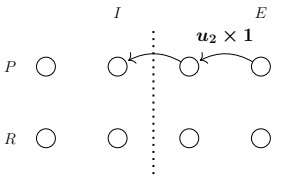
条件： $x_1 = 1$, $x_2 = 1$, $y = 0$, Tsetlin 自动机$_1^3$=E。因此，II型，$\neg x_1 = 0$, $C_3 = x_1 = 1$。



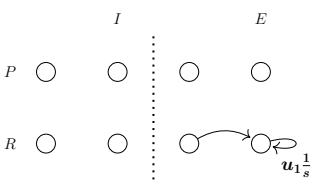
条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=E。因此，I 型，$\neg x_1 = 0$, $C_3 = x_1 = 1$.



条件： $x_1 = 1$, $x_2 = 1$, $y = 0$, Tsetlin 自动机$_1^3$=I。因此，II型,$\neg x_1 = 0$,$C_3 = x_1 = 1$.



条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=I。因此，I 型，$\neg x_1 = 0$, $C_3 = x_1 = 1$。

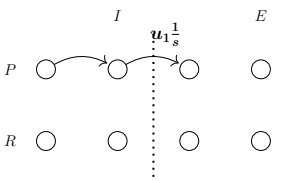


情况 4

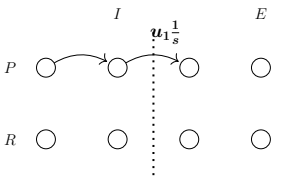
这里，Tsetlin 自动机$_3^3$ 被冻结为“包含”，Tsetlin 自动机$_4^3$ 也被冻结为“包含”。在这种情况下，该子句的输出始终为 0。

现在我们研究动作是“包含”的 Tsetlin 自动机$_1^3$。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=E。因此，I 型，$x_1 = 0$,$C_3 = 0$。

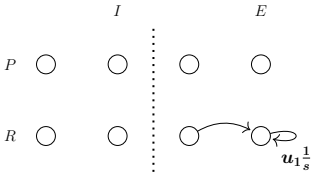


条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=I。因此，I 型，$x_1 = 0$, $C_3 = 0$

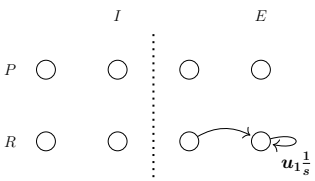


现在我们研究动作是“排除”的 Tsetlin 自动机$_1^3$。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_2^3$=E。因此，I 型，$x_1 = 0$,$C_3 = 0$。

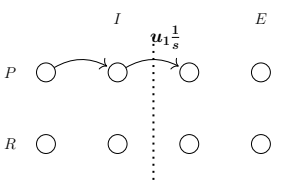


条件： $x_1 = 0$, $x_2 = 1$, $y = 0$, Tsetlin 自动机$_2^3$=I。因此，II型，$x_1 = 0$, $C_3 = 0$。

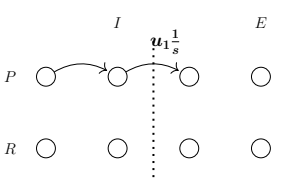


现在我们研究Tsetlin 自动机$_2^3$ **的“包含”动作**。

条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=E。因此，I 型，$\neg x_1 = 1$,$C_3 = 0$。

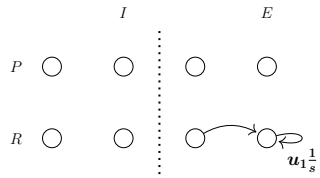


条件： $x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$_1^3$=I。因此，I 型，$\neg x_1 = 1$, $C_3 = 0$。

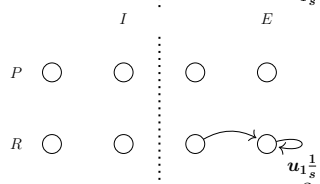


现在我们研究Tsetlin 自动机$_2^3$ **执行“排除”动作的情况**。

条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$^3_1$=E.
因此，Type I, $\neg x_1 = 1$, $C_3 = 0$.



条件：$x_1 = 0$, $x_2 = 1$, $y = 1$, Tsetlin 自动机$^3_1$=I.
因此，Type I, $\neg x_1 = 1$, $C_3 = 0$.



基于上述分析，我们可以展示在不同配置下 TA^3_3 和 TA^3_4 的 TA^3_1 和 TA^3_2 状态转移方向。

场景1： 研究 TA^3_1 = I 和 TA^3_2 = E。
情况 1： 我们可以看到，**情况 2：** 我们可以看到 $TA^3_1 \to$E $TA^3_1 \to$ E $TA^3_2 \to$E $TA^3_2 \to$ I、E**情况 3：** 我们可以看到，**情况 4：** 我们可以看到
$TA^3_1 \to$ E $TA^3_1 \to$ E
$TA^3_2 \to$ I,E $TA^3_2 \to$ E

场景2： 研究 TA^3_1 = I 和 TA^3_2 = I。
情形 1： 我们可以看到，**情况 2：** 我们可以看到 $TA^3_1 \to$E $TA^3_2 \to$E**ase：** 我们可以看到，**情况 4：** 我们可以看到 $TA^3_1 \to$ E $TA^3_1 \to$E $TA^3_2 \to$ E $TA^3_2 \to$E

情景 3： 研究 Tsetlin 自动机$^3_1$ = E 和 Tsetlin 自动机$^3_2$ = I。
情况 1： 我们可以看到 **情况 2：** 我们可以看到Tsetlin 自动机$^3_1 \to$ E Tsetlin 自动机$^3_2 \to$E**ase：** 我们可以看到 **情况 4：** 我们可以看到 Tsetlin 自动机$^3_1 \to$ I Tsetlin 自动机$^3_1 \to$E Tsetlin 自动机$^3_2 \to$ I Tsetlin 自动机$^3_2 \to$E

场景4： 研究 Tsetlin 自动机$^3_1$ = E 和 Tsetlin 自动机$^3_2$ = E。
情况 1： 我们可以看到 **情况 2：** 我们可以看到
$TA^3_1 \to$ I, E Tsetlin 自动机$^3_1 \to$E Tsetlin 自动机$^3_2 \to$I
$TA^3_2 \to$ E
C况： 我们可以看到**情况 4：** 我们可以看到
$TA^3_1 \to$ E $TA^3_1 \to$ E
$TA^3_2 \to$ I, E $TA^3_2 \to$ E

从上述转移可以得出，当 Tsetlin 自动机$^3_3$=I 和 Tsetlin 自动机$^3_4$=E 处于冻结状态时，状态 Tsetlin 自动机$^3_1$=E 和 Tsetlin 自动机$^3_2$=I 是吸收的。同样，当 Tsetlin 自动机$^3_3$ 和 Tsetlin 自动机$^3_4$ 均冻结在 Include 状态时，状态 Tsetlin 自动机$^3_1$=E 和 Tsetlin 自动机$^3_2$=E 也是吸收的。其他状态都不是吸收状态。