The Power of Hard Attention Transformers on Data Sequences: A Formal Language Theoretic Perspective

38th Annual Conference on Neural Information Processing Systems, Vancouver

Pascal Bergsträßer[1] Chris Köcher[2] Anthony Widjaja Lin[1,2] Georg Zetzsche[2]

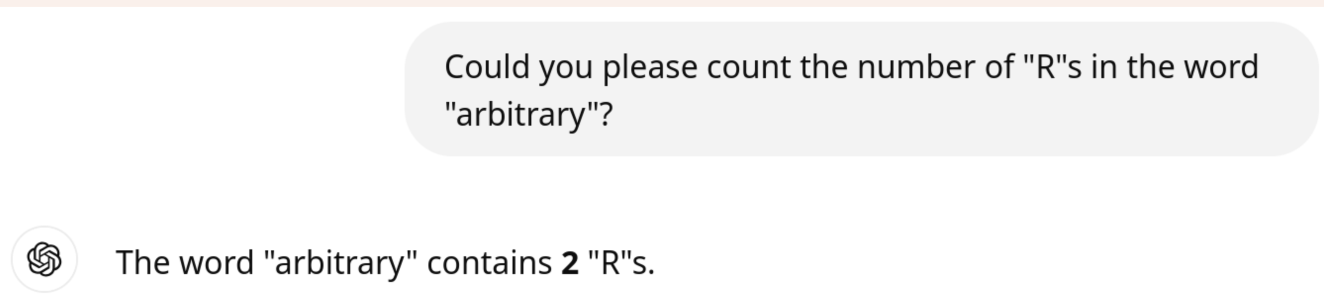
[1] Rheinland-Pfälzische Technische Universität, Kaiserslautern-Landau, Germany
[2] Max Planck Institute for Software Systems, Kaiserslautern, Germany

What are Transformers?

- Introduced in "Attention is all you need" [Vaswani et al. @ NeurIPS 2017]
- Basic model used in recent Large Language Models (like ChatGPT, Gemini, . . .)
- Applications in natural language processing, computer vision, language recognition, time series analysis, . . .

Verifying Transformers?

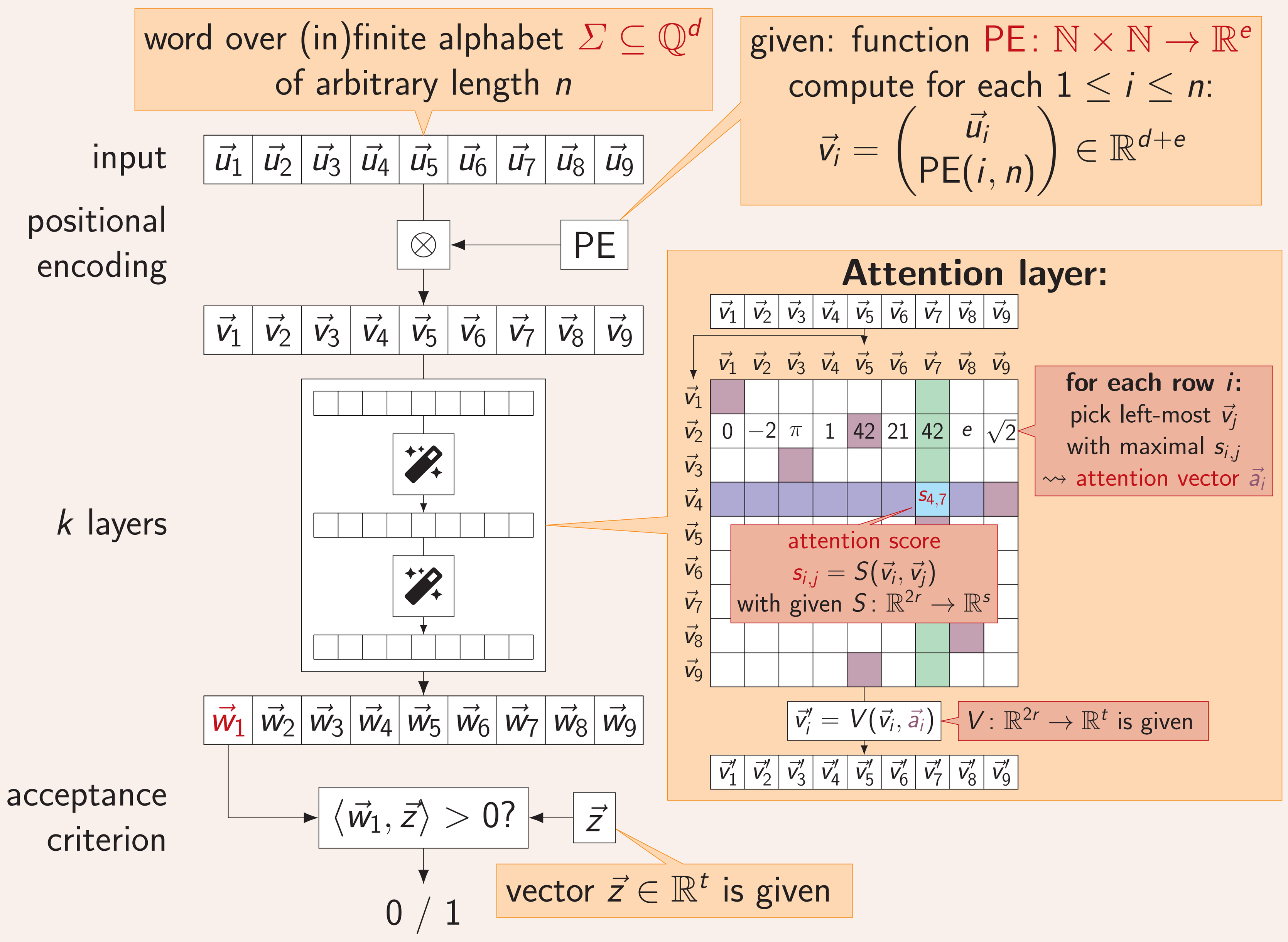
- Sometimes their output is just wrong:



(Spoiler: "arbitrary" contains **three** "R"s)

- Questions:
  1. What can be expressed via transformers?
     - e.g. Parity $= \{w \in \{a,b\}^* : |w|_a$ is even$\}$ is accepted by a transformer [Chiang & Cholak @ ACL 2022]
  2. What can transformers learn?
     - e.g. Parity is not trainable via known algorithms [Bhattamishra et al. @ EMNLP 2020]
  3. Can we verify the (in)correctness of a transformer? If yes, how?
- Already a lot of work done for transformers on words.
- Not (much) covered yet: transformers taking complex input data (like pictures, videos, voice, . . .)

Unique Hard Attention Transformers (UHAT)



UHAT$_{\text{fin}}$ = all languages over finite $\Sigma \subseteq \mathbb{Q}^d$ accepted by a UHAT
UHAT$_{\text{inf}}$ = all languages over infinite $\Sigma \subseteq \mathbb{Q}^d$ accepted by a UHAT

UHAT vs. Circuit Complexity

Circuit Complexity Classes AC^0 and TC^0

AC^0 = all languages accepted by family of circuits of
- constant depth,
- polynomial size, and
- Boolean gates with unbounded fan-in.

TC^0 extends AC^0 by majority gates.

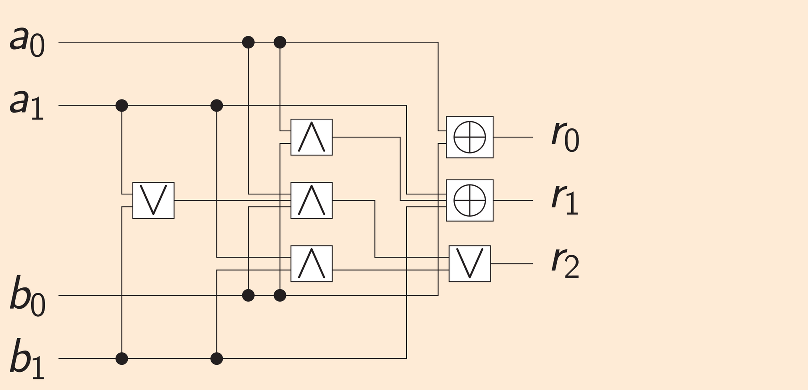


Figure: Circuit for $a_1a_0 + b_1b_0 = r_2r_1r_0$. In general, addition is in AC^0.

Well-known: AC$^0 \subset$ TC^0

Known Result

UHAT$_{\text{fin}} \subset$ AC^0
[Hao et al. @ TACL 2022; Barceló et al. @ ICLR 2024]

Theorem

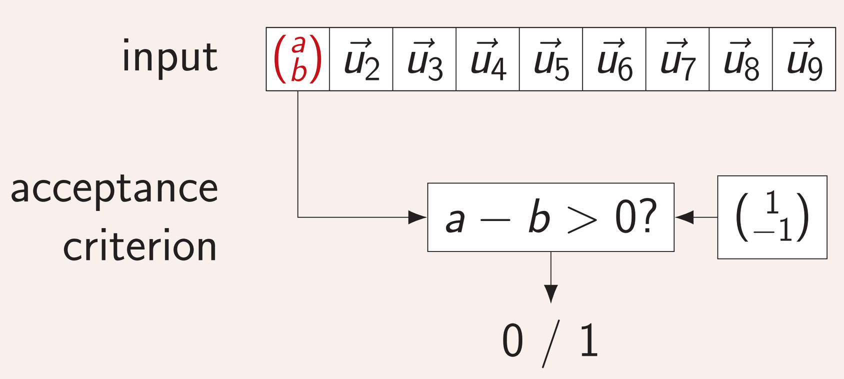
UHAT$_{\text{inf}} \subseteq$ TC^0

Theorem

$$\text{Compare} = \left\{\binom{a}{b} \cdot w \in (\mathbb{Q}^2)^* \;\middle|\; a > b\right\}$$

is in UHAT$_{\text{inf}} \setminus$ AC^0.

Proof. The following UHAT (without positional encoding) and 0 layers accepts Compare:



□

UHAT vs. Regular Languages

Known Result

UHAT$_{\text{fin}}$ (*without positional encoding* but with masking) = StarFree ⊂ Regular [Yang et al. @ NeurIPS 2024]
(StarFree = languages constructed from finite languages using Boolean operations and concatenation)

Symbolic Automata and Regular Data Languages

- Symbolic automata = NFA with arithmetic constraints as edge labels

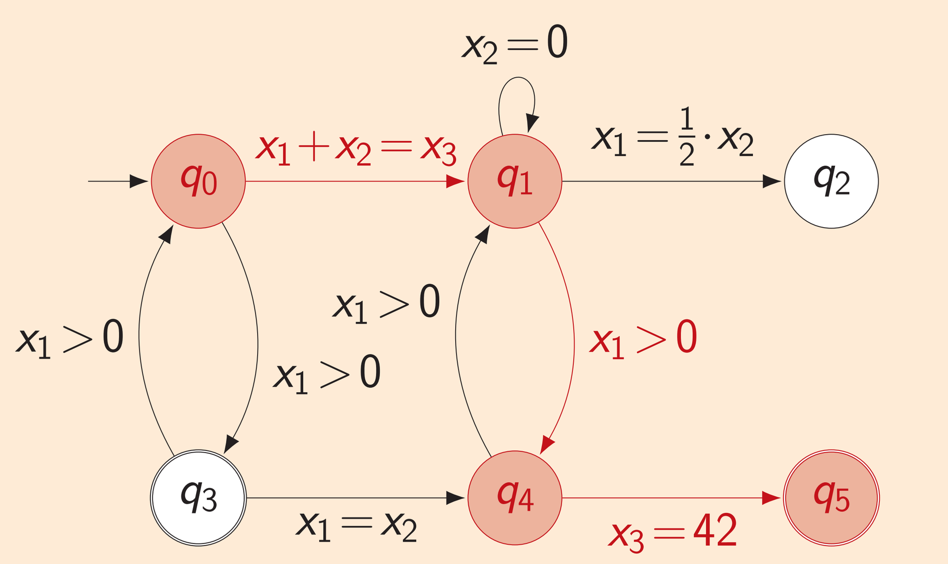


Figure: A symbolic automaton. Colored edges mark an accepting run of $\left(\begin{pmatrix}1\\2\\3\end{pmatrix}, \begin{pmatrix}0.1\\0\\0\end{pmatrix}, \begin{pmatrix}10.5\\21\\42\end{pmatrix}\right)$

- $p \xrightarrow{\vec{x}} q$ if there is a transition (p, ϕ, q) with $\vec{x} \in [\![\phi]\!]$
- $(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n) \in L(\mathfrak{A}) \iff I \xrightarrow{\vec{x}_1}\xrightarrow{\vec{x}_2} \cdots \xrightarrow{\vec{x}_n} F$

Theorem

Define Double $\subseteq \mathbb{Q}^*$:

$$\{(x_1, x_2, \ldots, x_n) \in \mathbb{Q}^* \mid \forall 1 \le i < n \colon 2x_i < x_{i+1}\}.$$

Double ∉ Regular, but Double ∈ UHAT$_{\text{inf}}$ (without positional encoding but with masking)

Proof. Non-regularity is shown by Pumping Lemma. Double is accepted by a UHAT with two layers:

1. For each position i choose the $j > i$ maximizing
$$2x_i - x_j$$
and let y_i be this maximal value.
2. Check whether all y_i's are non-positive. □

UHAT vs. Logic

Linear Temporal Logic (LTL)

LTL describes words via following syntax and semantics:

		a	b	b	a	c	a	a	b
Atoms	a:	t	f	f	t	f	t	t	f
Boolean ops.	$a \vee b$:	t	t	t	t	f	t	t	t
Ne**X**t op.	X a:	f	f	t	f	t	t	f	f
Until op.	a U b:	t	t	t	f	f	t	t	t
Finally op.	F c:	t	t	t	t	t	f	f	f
Globally op.	G($a \vee b$):	f	f	f	f	f	t	t	t

$L(\phi) = \{w \in \Sigma^* \mid \phi$ holds in first position of $w\}$
Well-known: LTL = StarFree [Kamp 1968]
LTL(Mon) extends LTL by positional predicates.

Known Result

LTL(Mon) ⊆ UHAT$_{\text{fin}}$ [Barceló et al. @ ICLR 2024]

Locally Testable LTL (LTLTL)

LTLTL extends LTL(Mon) to alphabet $\Sigma = \mathbb{Q}^d$ by adding arithmetic constraints of the form

$$\langle (\vec{x}_i, \vec{x}_{i+1}, \ldots, \vec{x}_{i+k}), \vec{a}\rangle > b.$$

Theorem

LTLTL ⊆ UHAT$_{\text{inf}}$

Example: 7-day Simple Moving Average

- Check for an "uptrend" in a time series
- Describes time sequences where the value at each time t is above the average of the week ending at t.
- G $\left(\mathrm{X}^6\,\text{true} \to 7x_{i+6} > x_i + x_{i+1} + \cdots + x_{i+6}\right)$.