Length Generalization Bounds for Transformers

Andy Yang[1] Pascal Bergsträßer[2] Georg Zetzsche[3] David Chiang[1] Anthony Lin[2,3]

June 2, 2026

[1]University of Notre Dame

[2]RPTU Kaiserslautern-Landau

[3]Max Planck Institute for Software Systems

Length generalization

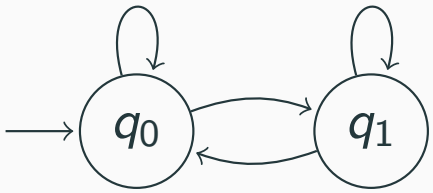
If we train on length up to N . . .

Will we succeed on lengths up to $2N$ . . .?

And how large do we have to make N . . .?

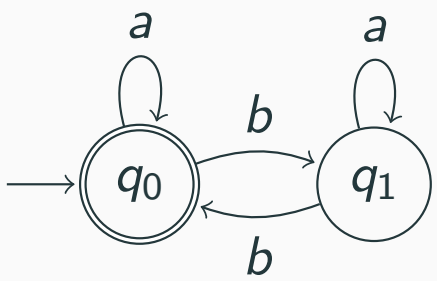
Length generalization bounds

Chen et al. 2025 considered the question, what is the minimum input length needed to guarantee length-generalization? In the case of 2-state automata:



ϵ	accept
a	accept
b	reject
aa	accept
ab	reject
ba	reject
bb	accept

After seeing all strings of up to length 2, we can always recover the ground truth 2-state automaton.



C-RASP

A programming language defined by Yang & Chiang 2024 based on RASP invented by Weiss et al. 2021.

Dyck Language

$C_a(i) := \#\,[j \leq i]\ a(j)$	The number of a up to position i	(1)
$C_b(i) := \#\,[j \leq i]\ b(j)$	The number of b up to position i	(2)
$V(i) := C_a(i) < C_b(i)$	*Violation*: there are more b than a	(3)
$C_V(i) := \#\,[j \leq i]\ V(j)$	The number of *Violations*	(4)
$M(i) := C_V(i) = 0$	*Matched* i.e. zero *Violations*	(5)
$B(i) := C_a(i) = C_b(i)$	*Balanced* i.e. same number of a and b	(6)
$D(i) := M(i) \wedge B(i)$	String is *Matched* and *Balanced*	(7)

Yang et al. 2024 showed that transformers can simulate C-RASP programs.

Previous bounds

Chen et al. 2025 showed length complexity bounds for depth-1 and depth-2 fragments of C-RASP:

Class	Complexity Measure	Length Complexity
C-RASP^{1,T}	precision T	$O(T^2)$
C-RASP^{2,K,T}	precision T and # of heads K	$O(T^{O(K)})$

C-RASP: **uncomputable bounds**

For any Diophantine system of equations...

$$\mathcal{D} := x^4 + y^4 + z^4 = w^4$$

There is a C-RASP program $\phi_{\mathcal{D}}$ such that $\mathcal{D}$ has a solution ...

$$x = 95800, y = 217519, z = 414560, w = 422481$$

whenever $\phi_{\mathcal{D}}$ accepts some word...

$$\cdots \sigma_x^{95800} \cdots \sigma_y^{217519} \cdots \sigma_z^{414560} \cdots \sigma_2^{422481} \cdots \models \phi$$

We show that the undecidability of Diophantine equations implies **uncomputable length generalization bounds for** C-RASP.

C-RASP$_+$: **exponential bounds**

For any C-RASP$_+$ program ϕ we obtain a TL[$\Diamond$] program ϕ' with an exponential blowup in size

$$\overleftarrow{\#}[\psi] \geq C \mapsto \underbrace{\Diamond(\psi \wedge \Diamond(\psi \wedge \ldots \Diamond(\psi \wedge \Diamond\psi)\ldots))}_{C \text{ many}} \vee$$

$$\underbrace{\psi \wedge \Diamond(\psi \wedge \Diamond(\psi \wedge \ldots \Diamond(\psi \wedge \Diamond\psi)\ldots)}_{C-1 \text{ many}}$$

Using the small model property of TL[$\Diamond$] if ϕ' accepts any string it must accept some string of length polyonmial in $|\phi'|$ (and thus exponential in $|\phi|$)

Summary

- We show C-RASP programs (and transformers) have uncomputable length complexity via a reduction from Diophantine equations
- We show the positive fragment C-RASP$_+$ (and finite precision transformers) have exponential length complexity

Class	Complexity Measure	Length Complexity
C-RASP^{1,T}	precision T	$O(T^2)$
C-RASP^{2,K,T}	precision T and # of heads K	$O(T^{O(K)})$
C-RASP	program size S	uncomputable
C-RASP$_+$	program size S	$O(2^{\text{poly}(S)})$