On the Practical Computational Power of Finite Precision RNNs for Language Recognition

Gail Weiss, Yoav Goldberg, Eran Yahav

GRU < LSTM (!?)

Supported by European Union's Seventh Framework Programme (FP7) under grant agreement no. 615688 (PRIME)

Current State

- RNNs are everywhere
- We don’t know too much about the differences between them:
    - Gated RNNs are shown to train better, beyond that:
    - “RNNs are Turing Complete”?

Turing Complete?

On the Computational Power of Neural Nets*

HAVA T. SIEGELMANN†

Department of Information Systems Engineering, Technion, Haifa 32000, Israel

AND

EDUARDO D. SONTAG‡

Department of Mathematics, Rutgers University, New Brunswick, New Jersey 08903

Received February 4, 1992; revised May 24, 1993

Turing Complete?

1993 Proof:

1. **Requires Infinite Precision:**
   Uses stack(s), maintained in certain dimension(s)
   Zeros are pushed using division (using g = g/4 + 1/4)
   *In 32 bits, this reaches the limit after **15** pushes*

2. **Requires Infinite Time:**
   Allows processing steps beyond reading input
   (Not the standard use case!)

unreasonable assumptions!

Turing Complete?

1993 Proof:

1. Requires Infinite Precision:
   Uses stack(s), maintained in certain dimension(s)
   Zeros are pushed using division (using g = g/4 + 1/4)
   *In 32 bits, this reaches the limit after **15** pushes*

2. Requires Infinite Time:
   Allows processing steps beyond reading input
   (Not the standard use case!)

TURING TARPIT!

unreasonable assumptions!

What happens on real hardware and real use-cases?

Real Use

- Gated architectures have the best performance
  - LSTM and GRU are most popular
  - Of these, the choice between them is unclear

Main Result

We accept all RNN types can simulate DFAs

We show that LSTMs and IRNNs can also count

And that the GRU and SRNN cannot

Power of Counting

Practical

In NMT:
LSTM better at capturing target length

Power of Counting

Practical

In NMT:
LSTM better at capturing target length

Theoretical

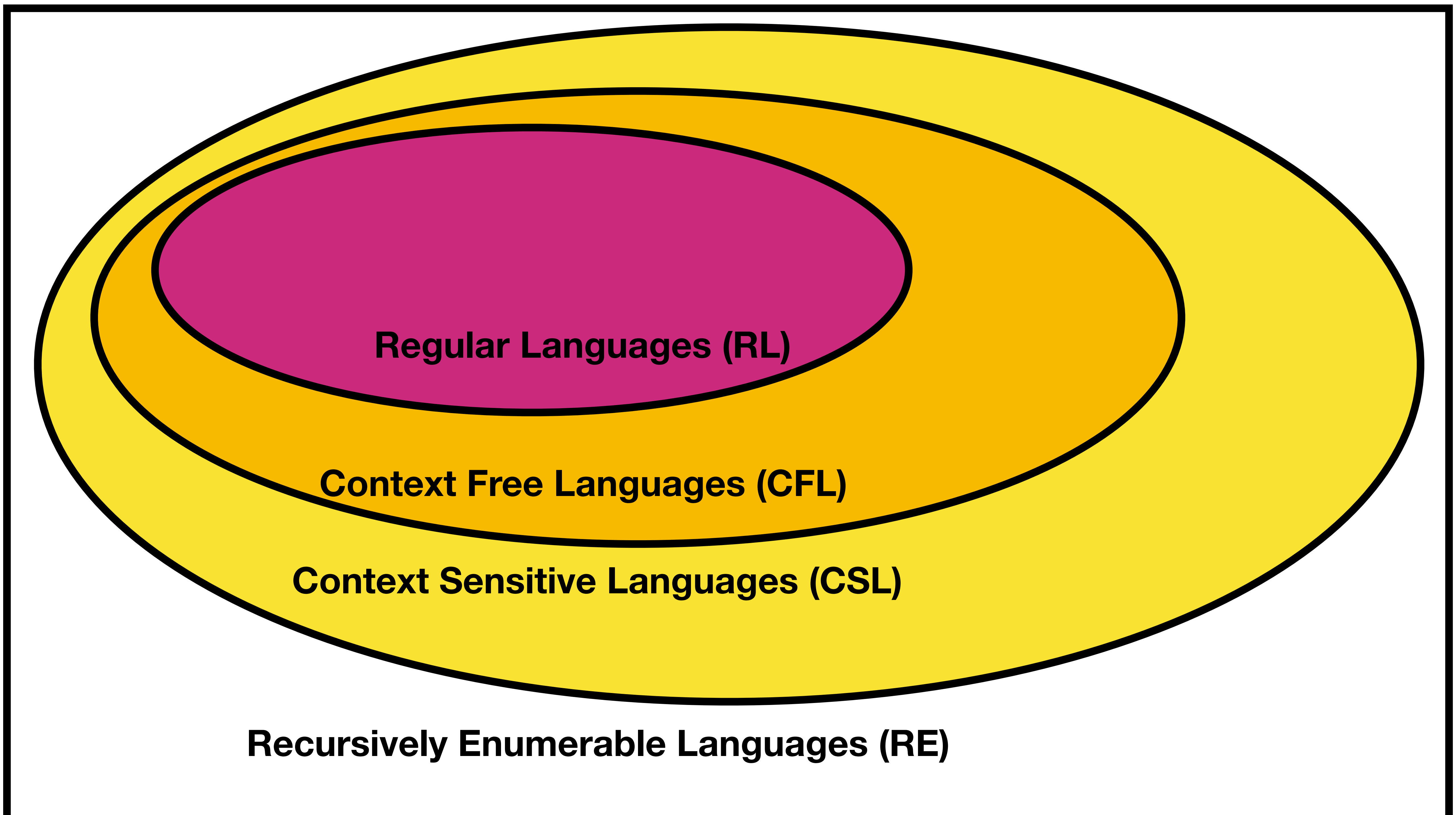
Finite State Machines vs **Counter Machines**

K-Counter Machines (SKCMs)

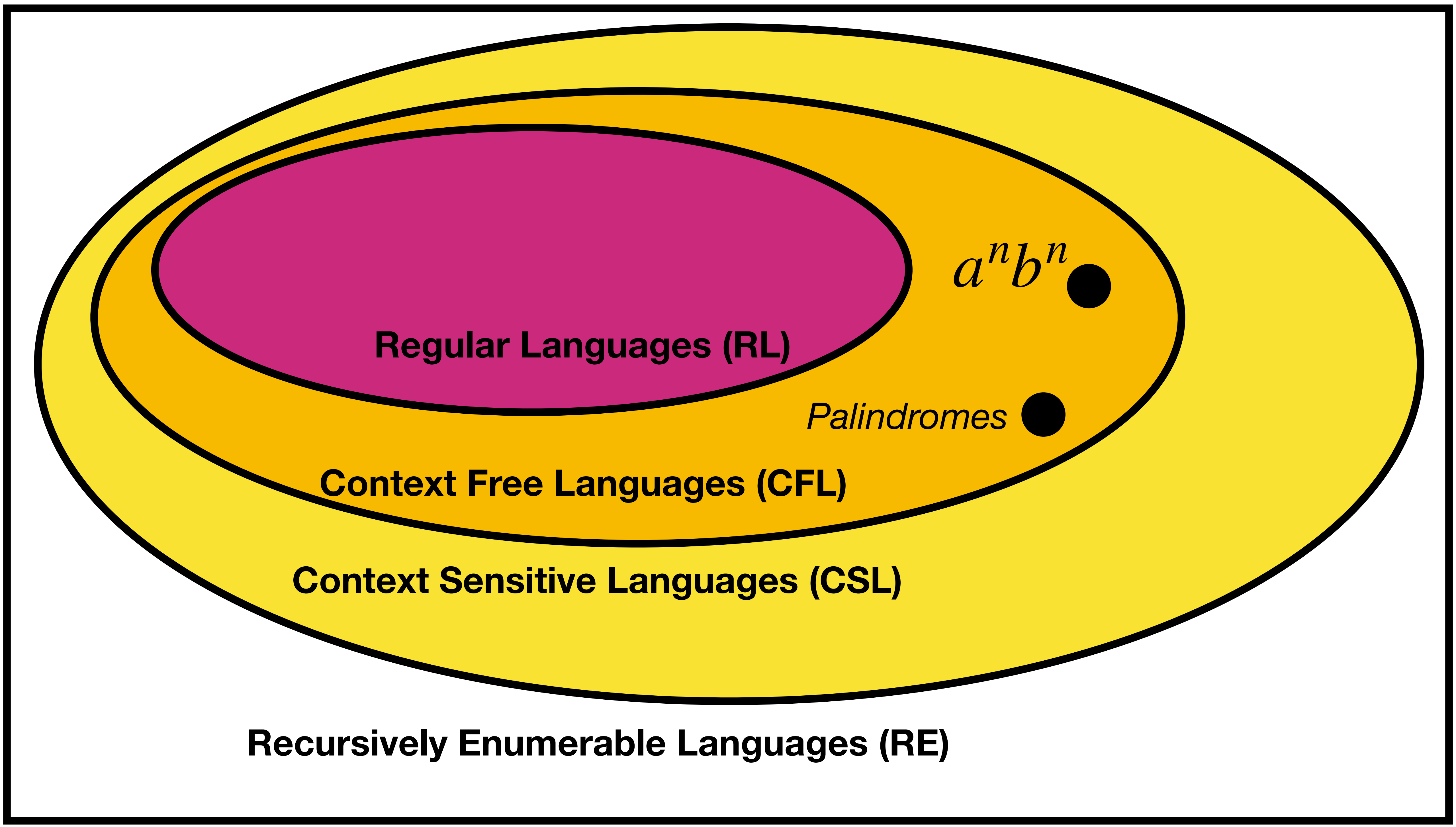
Fischer, Meyer, Rosenberg - 1968

- Similar to finite automata, but also maintain k *counters*
- A *counter* has 4 operations: inc/dec by one, do nothing, reset
- Counters are observed by comparison to zero

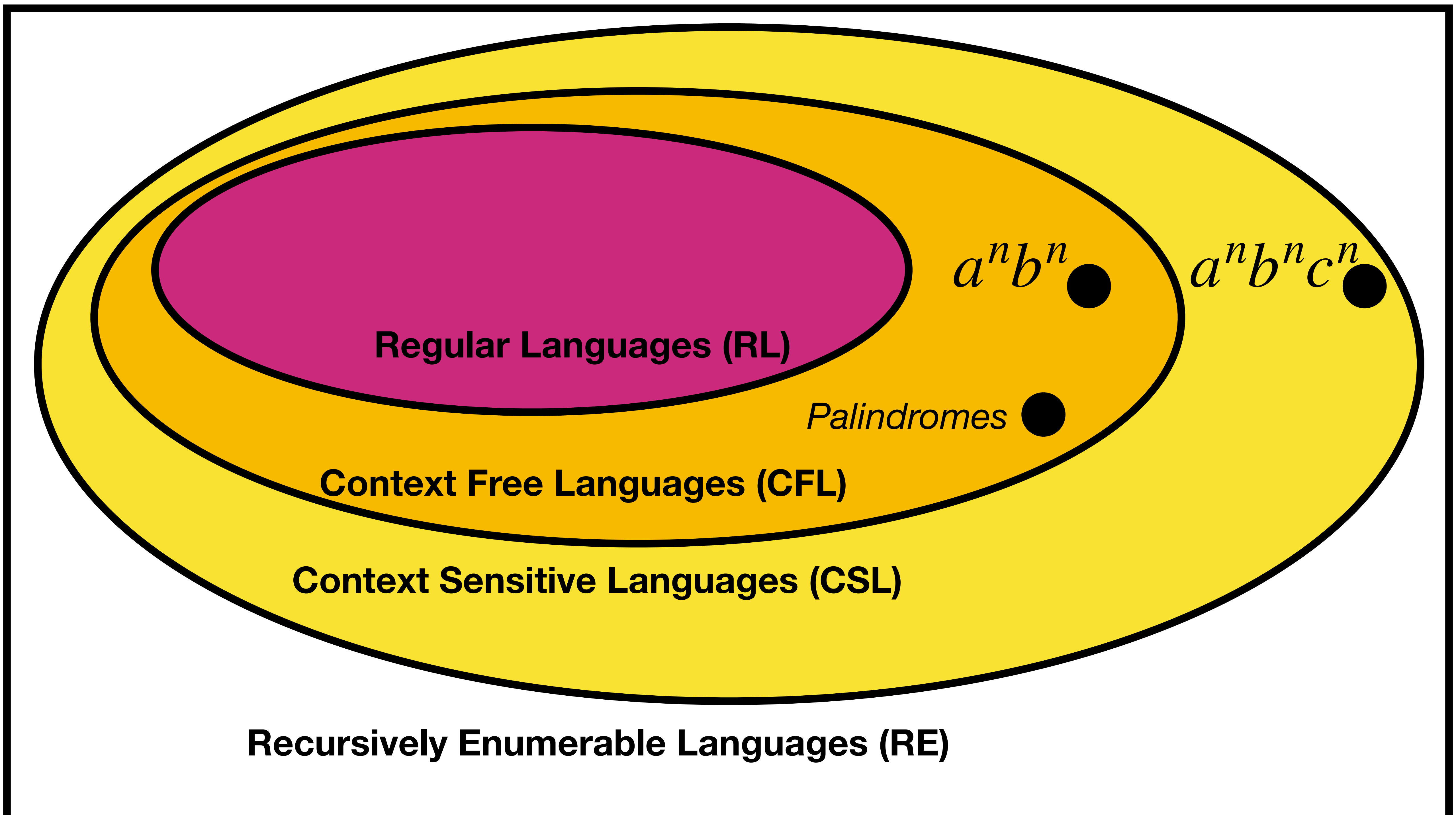
Counting Machines and Chomsky Hierarchy



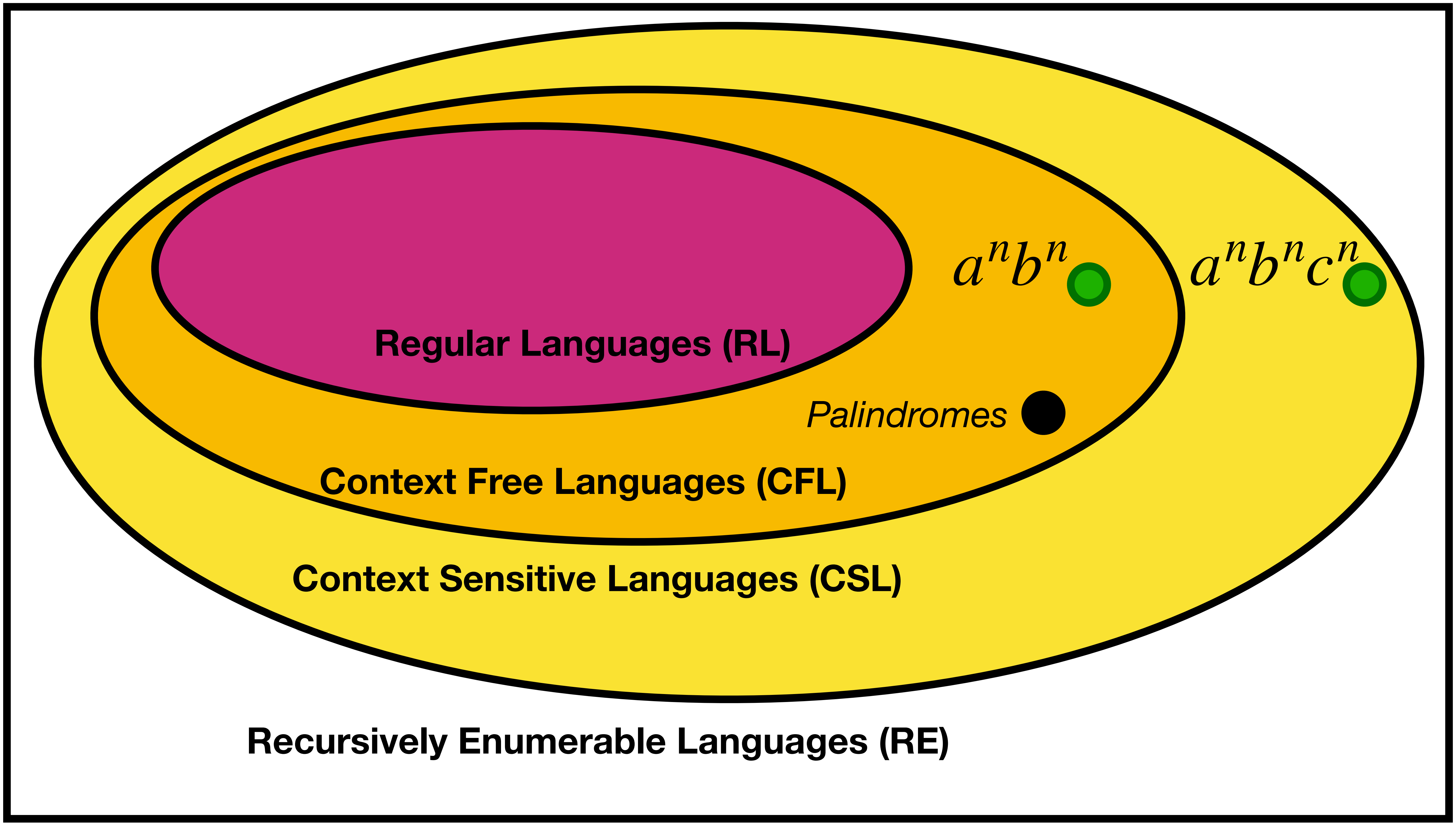
Chomsky Hierarchy and SKCMs



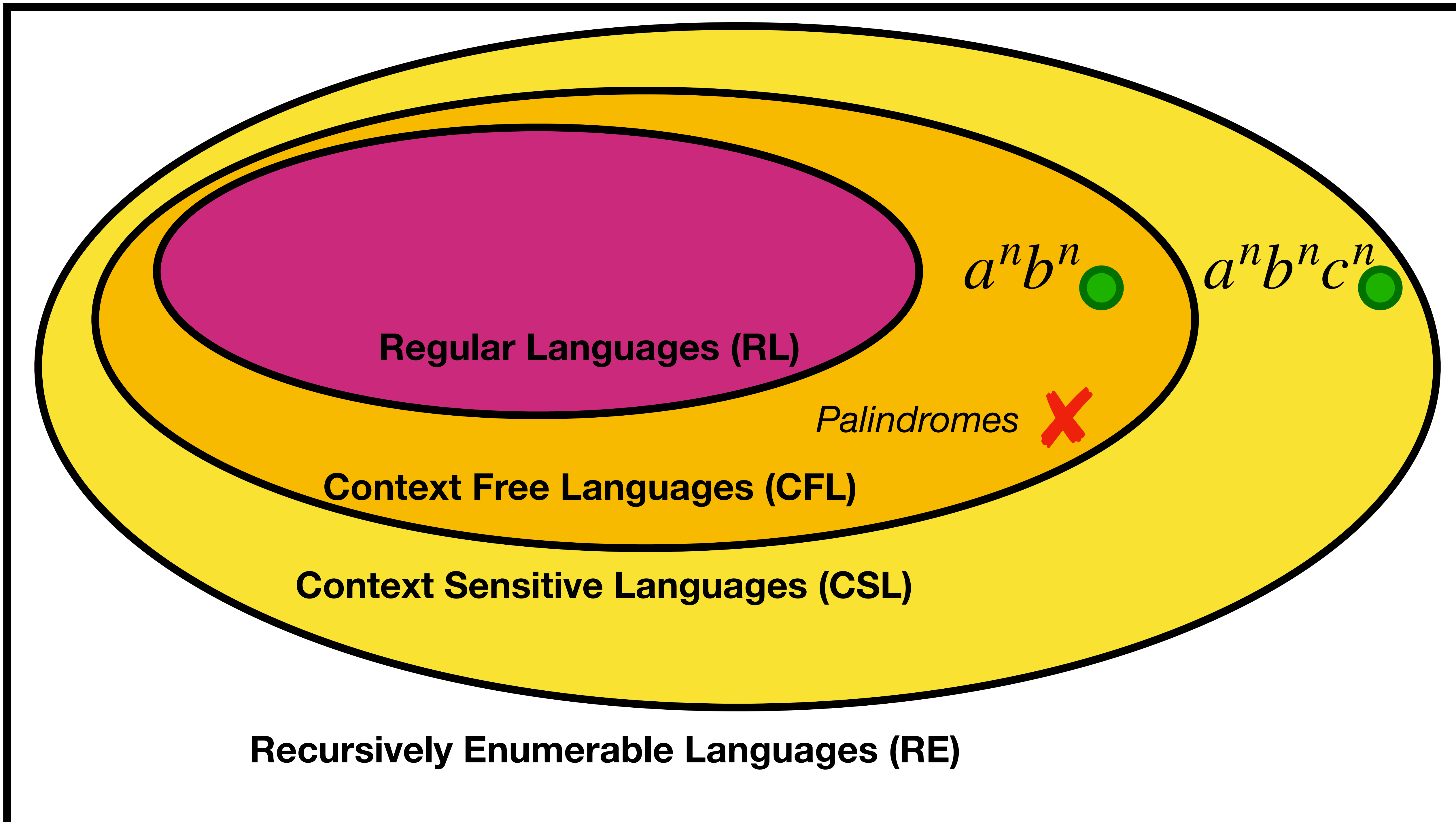
Chomsky Hierarchy and SKCMs



Chomsky Hierarchy and SKCMs

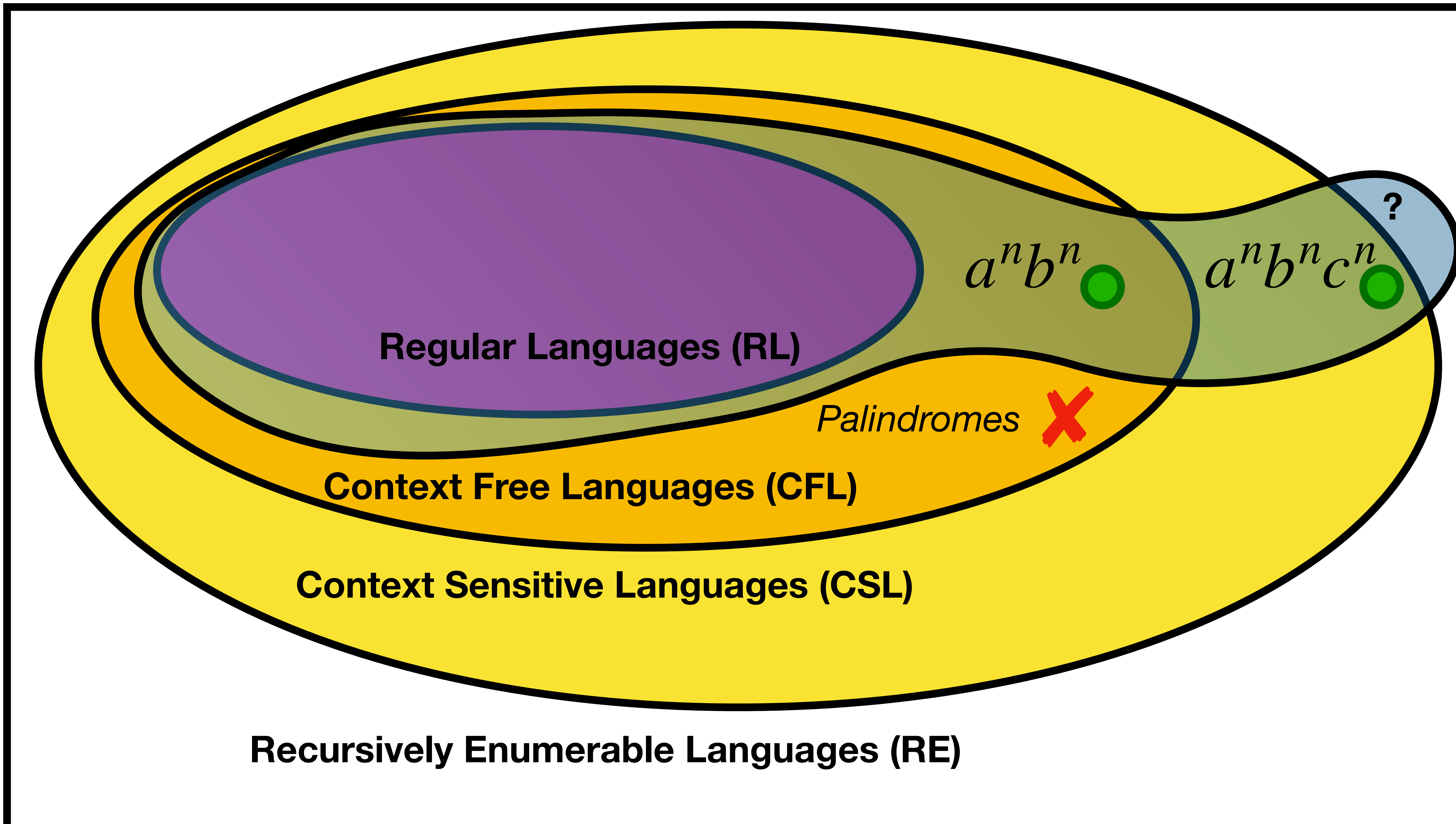


Chomsky Hierarchy and SKCMs



Chomsky Hierarchy and SKCMs

SKCMs cross the Chomsky Hierarchy!



Summary so Far

- Counters give additional formal power
- We claimed that LSTM can count and GRU cannot

Summary so Far

- Counters give additional formal power
- We claimed that LSTM can count and GRU cannot
- **Let’s see why**

Popular Architectures

GRU

$$z_t = \sigma(W^z x_t + U^z h_{t-1} + b^z)$$
$$r_t = \sigma(W^r x_t + U^r h_{t-1} + b^r)$$
$$\tilde{h}_t = \tanh(W^h x_t + U^h(r_t \circ h_{t-1}) + b^h)$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

LSTM

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f)$$
$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i)$$
$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o)$$
$$\tilde{c}_t = \tanh(W^c x_t + U^c h_{t-1} + b^c)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$
$$h_t = o_t \circ g(c_t)$$

Popular Architectures

GRU

$$z_t = \sigma(W^z x_t + U^z h_{t-1} + b^z)$$
$$r_t = \sigma(W^r x_t + U^r h_{t-1} + b^r)$$
$$\tilde{h}_t = \tanh(W^h x_t + U^h (r_t \circ h_{t-1}) + b^h)$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

LSTM

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f)$$
$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i)$$
$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o)$$
$$\tilde{c}_t = \tanh(W^c x_t + U^c h_{t-1} + b^c)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$
$$h_t = o_t \circ g(c_t)$$

gates

candidate vectors

update functions

Popular Architectures

GRU

$$z_t \in (0,1)$$
$$r_t \in (0,1)$$
$$\tilde{h}_t = \tanh(W^h x_t + U^h(r_t \circ h_{t-1}) + b^h)$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

LSTM

$$f_t \in (0,1)$$
$$i_t \in (0,1)$$
$$o_t \in (0,1)$$
$$\tilde{c}_t = \tanh(W^c x_t + U^c h_{t-1} + b^c)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$
$$h_t = o_t \circ g(c_t)$$

gates

candidate vectors

update functions

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in (0,1)$

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$

LSTM

$f_t \in (0,1)$

$i_t \in (0,1)$

$o_t \in (0,1)$

$\tilde{c}_t \in (-1,1)$

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

$h_t = o_t \circ g(c_t)$

gates

candidate vectors

update functions

Popular Architectures

GRU

$$z_t \in (0,1)$$
$$r_t \in (0,1)$$
$$\tilde{h}_t \in (-1,1)$$
$$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$$

LSTM

$$f_t \in (0,1)$$
$$i_t \in (0,1)$$
$$o_t \in (0,1)$$
$$\tilde{c}_t \in (-1,1)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$
$$h_t = o_t \circ g(c_t)$$

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in (0,1)$

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$

LSTM

$f_t \in (0,1)$

$i_t \in (0,1)$

$o_t \in (0,1)$

$\tilde{c}_t \in (-1,1)$

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

$h_t = o_t \circ g(c_t)$

Interpolation

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \in (0,1)$

$i_t \in (0,1)$

$o_t \in (0,1)$

$\tilde{c}_t \in (-1,1)$

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

$h_t = o_t \circ g(c_t)$

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \in (0,1)$

$i_t \in (0,1)$

$o_t \in (0,1)$

$\tilde{c}_t \in (-1,1)$

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

$h_t = o_t \circ g(c_t)$

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \in (0,1)$

$i_t \in (0,1)$

$o_t \in (0,1)$

$\tilde{c}_t \in (-1,1)$

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

$h_t = o_t \circ g(c_t)$

Addition

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \approx 1$

$i_t \approx 1$

$o_t \in (0,1)$

$\tilde{c}_t \in (-1,1)$

$c_t \approx c_{t-1} + \tilde{c}_t$

$h_t = o_t \circ g(c_t)$

Addition

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1-z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \approx 1$

$i_t \approx 1$

$o_t \in (0,1)$

$\tilde{c}_t \approx 1$

$c_t \approx c_{t-1} + 1$

$h_t = o_t \circ g(c_t)$

Increase by 1

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \approx 1$

$i_t \approx 1$

$o_t \in (0,1)$

$\tilde{c}_t \approx -1$

$c_t \approx c_{t-1} - 1$

$h_t = o_t \circ g(c_t)$

Decrease by 1

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \approx 1$

$i_t \approx 0$

$o_t \in (0,1)$

$\tilde{c}_t \in (-1,1)$

$c_t \approx c_{t-1}$

$h_t = o_t \circ g(c_t)$

Do Nothing

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1-z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \approx 0$

$i_t \approx 0$

$o_t \in (0,1)$

$\tilde{c}_t \in (-1,1)$

$c_t \approx 0$

$h_t = o_t \circ g(c_t)$

Reset

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

Popular Architectures

GRU

$z_t \in (0,1)$

$r_t \in$

Bounded!

$\tilde{h}_t \in (-1,1)$

$h_t = z_t \circ h_{t-1} + (1 - z) \circ \tilde{h}_t$

Interpolation

LSTM

$f_t \approx 0$

$i_t \approx 0$

Can Count!

$o_t \in ($

$\tilde{c}_t \in (-1,1)$

$c_t \approx 0$

$h_t = o_t \circ g(c_t)$

Reset

$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

Other Architectures

SRNN

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$

IRNN

$$h_t = \max(0, W_h x_t + U_h h_{t-1} + b_h)$$

Other Architectures

SRNN

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \in (0,1)$$

Bounded!

IRNN

$$h_t = \max(0, W_h x_t + U_h h_{t-1} + b_h)$$

Other Architectures

SRNN

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \in (0,1)$$

Bounded!

IRNN

$$h_t = \underbrace{\max(0,}\, W_h x_t + U_h h_{t-1} + b_h)$$

keep/reset

+0 / +1

(subtraction in parallel, also increasing, counter)

Other Architectures

SRNN

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \in (0,1)$$

Bounded!

IRNN

$$h_t = \max(0, W_h x_t + U_h h_{t-1} + b_h)$$

Can Count!

+0 / +1

keep/reset

(subtraction in parallel, also increasing, counter)

So:

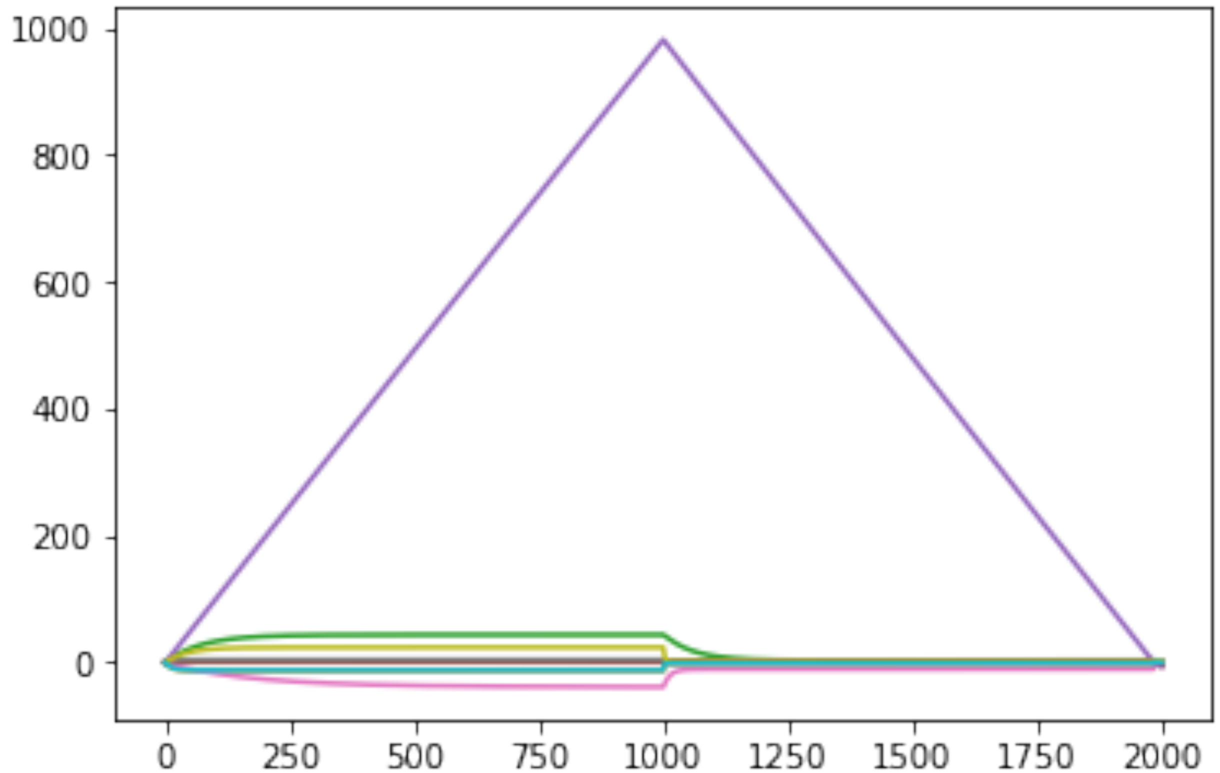
- LSTM can count!
- GRU cannot
- Counting gives greater computational power

Empirically

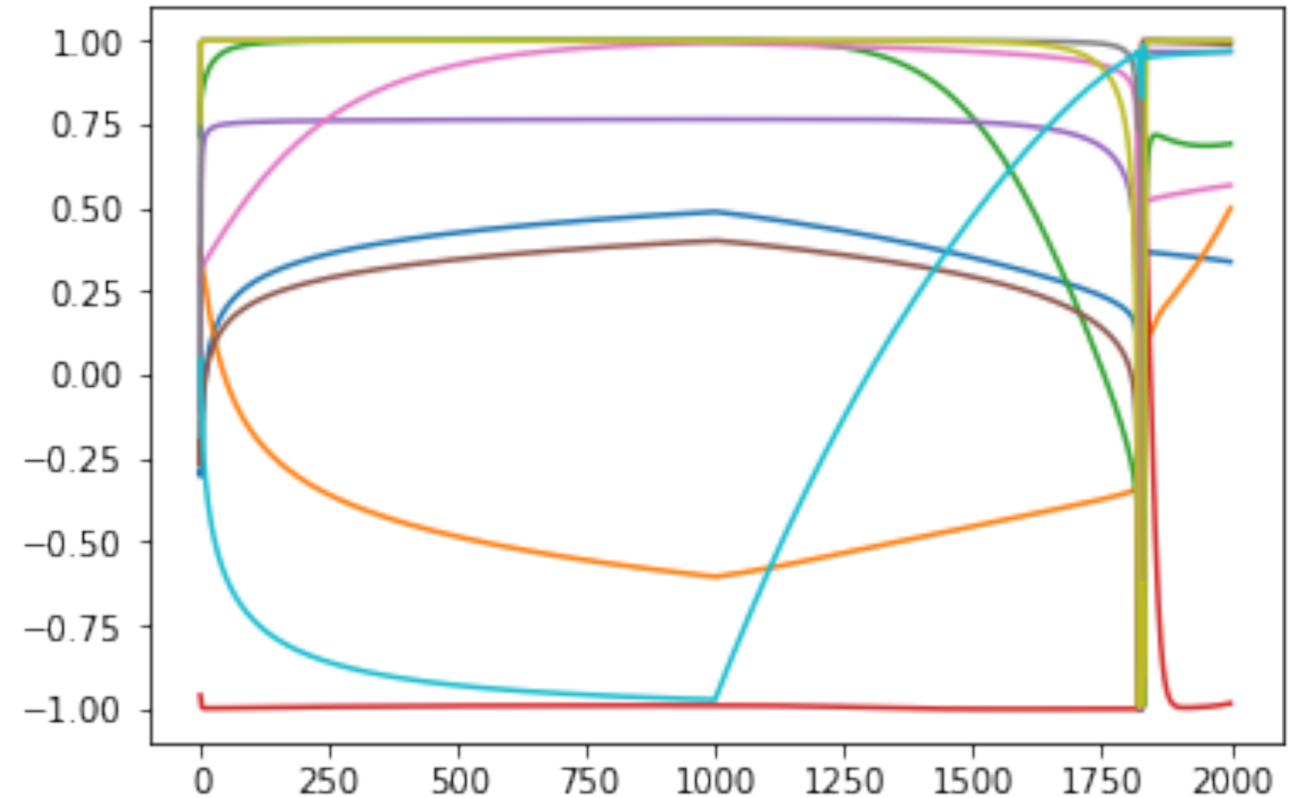
Trained $a^n b^n$**,** (on positive examples up to length 100)

Activations on $a^{1000} b^{1000}$ **:**

LSTM



GRU

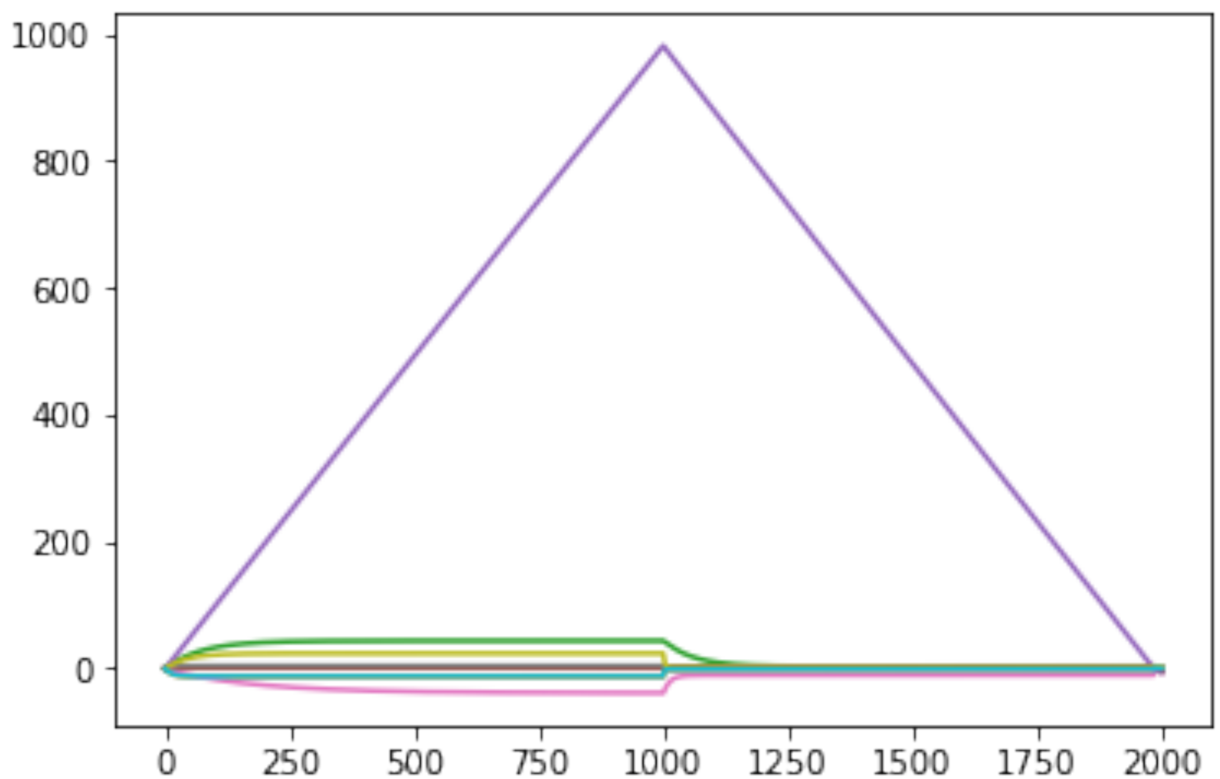


Empirically

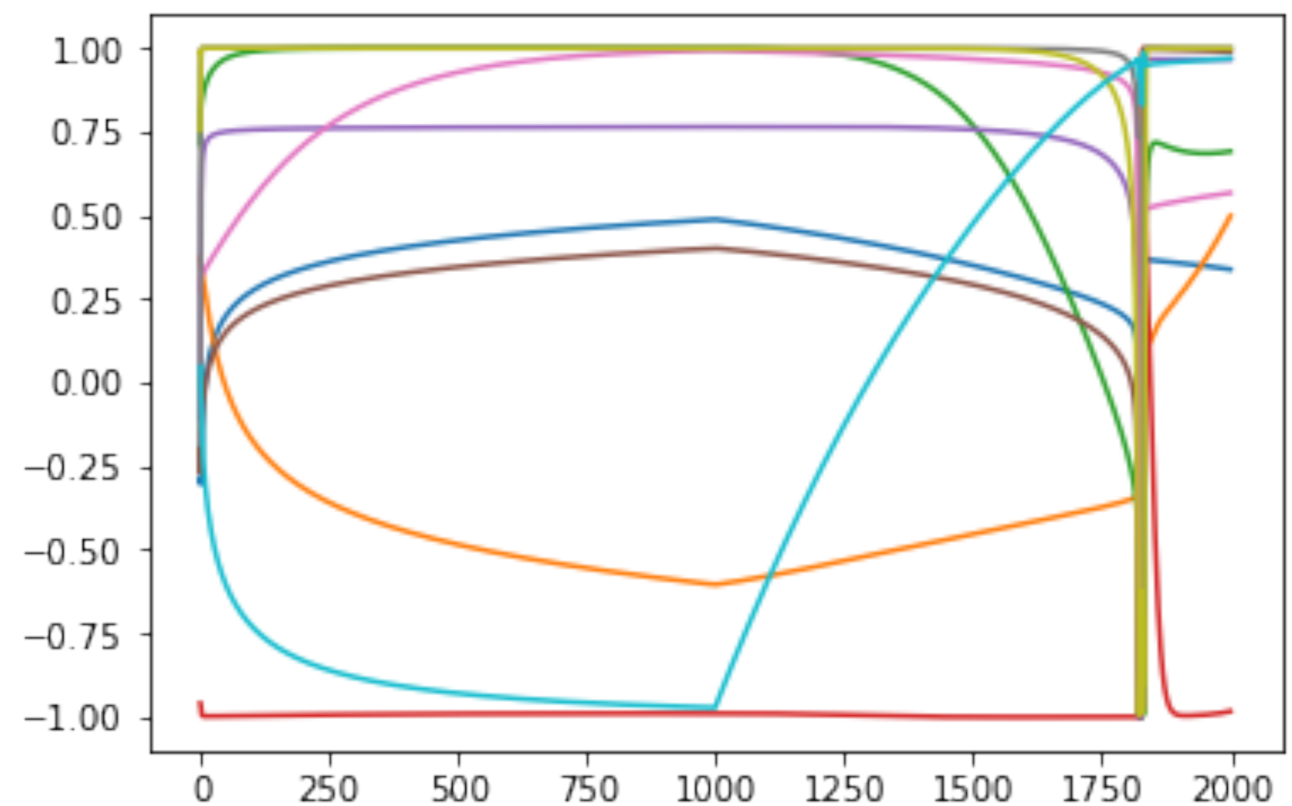
Trained $a^n b^n$**,** (on positive examples up to length 100)

Activations on $a^{1000} b^{1000}$ **:**

LSTM



GRU



GRU:

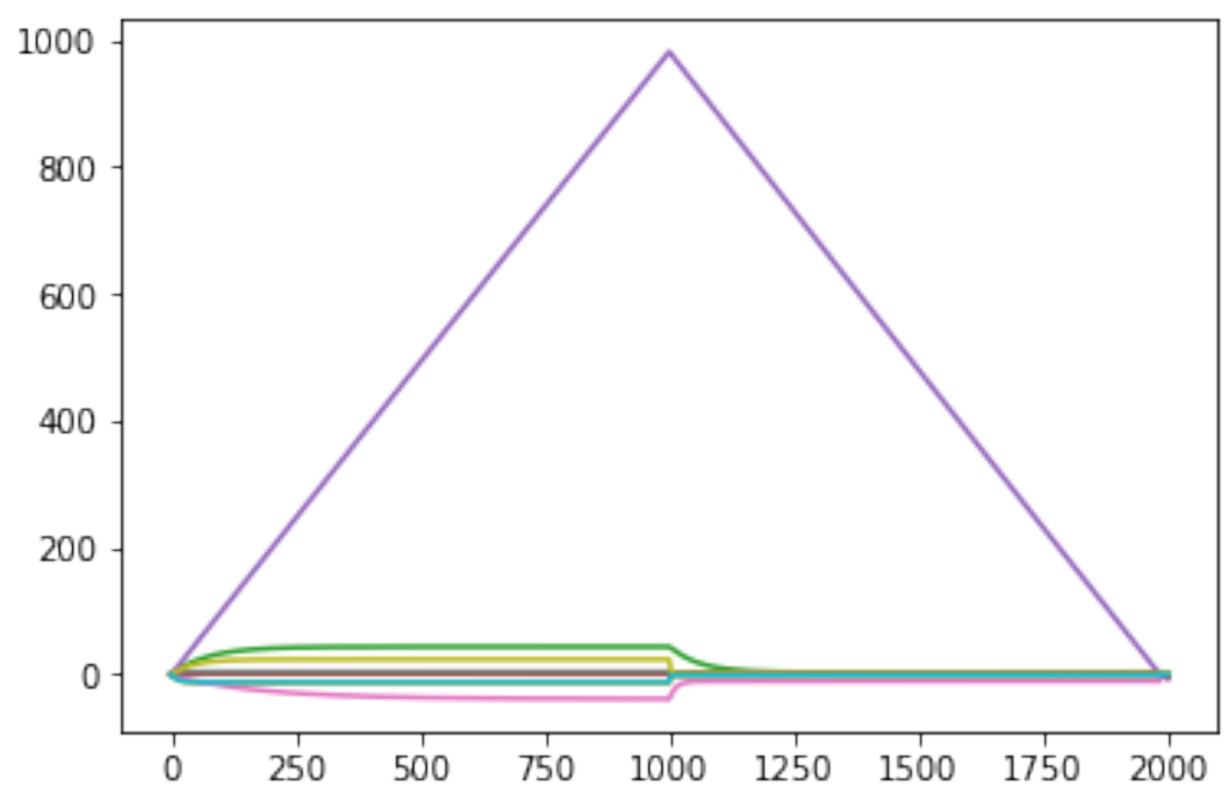
- **Took much longer to train**

Empirically

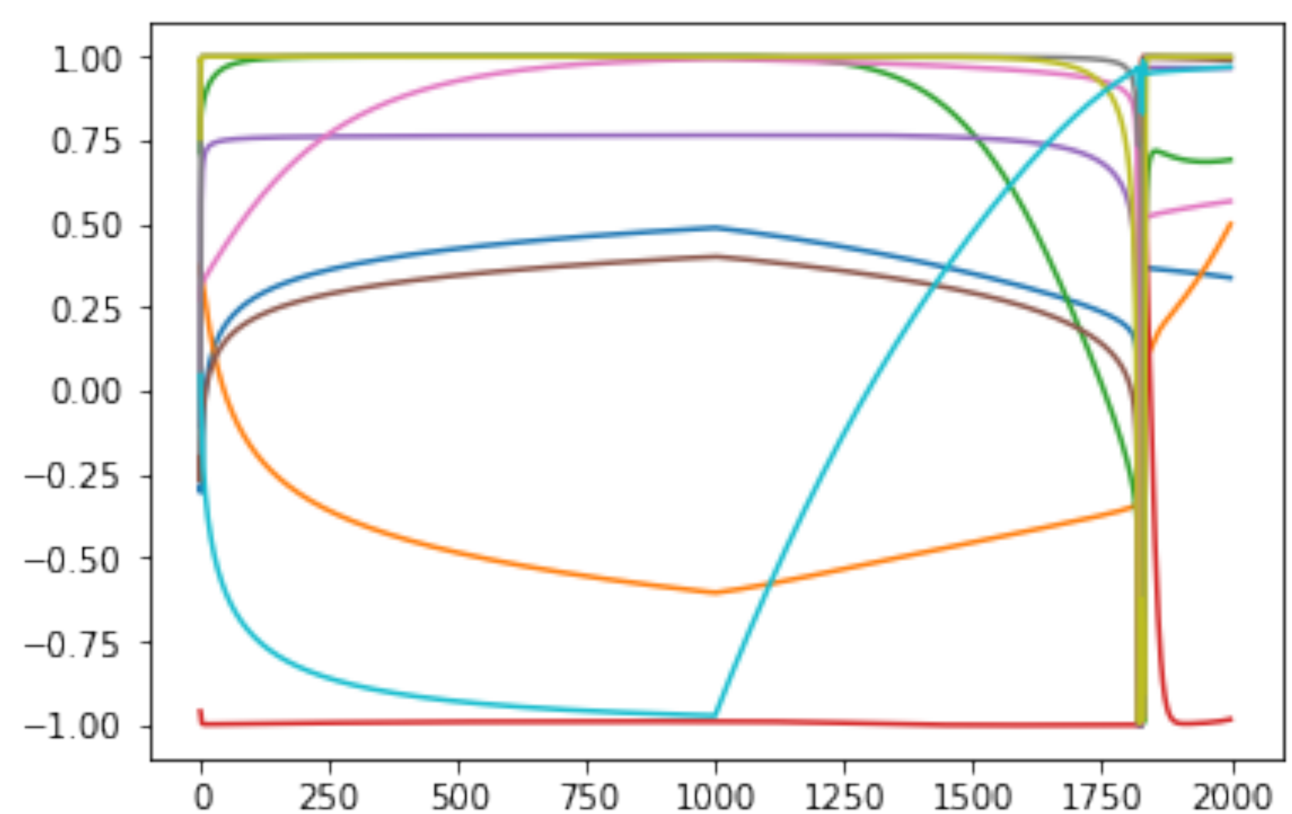
Trained $a^n b^n$**,** (on positive examples up to length 100)

Activations on $a^{1000} b^{1000}$ **:**

LSTM



GRU



GRU:

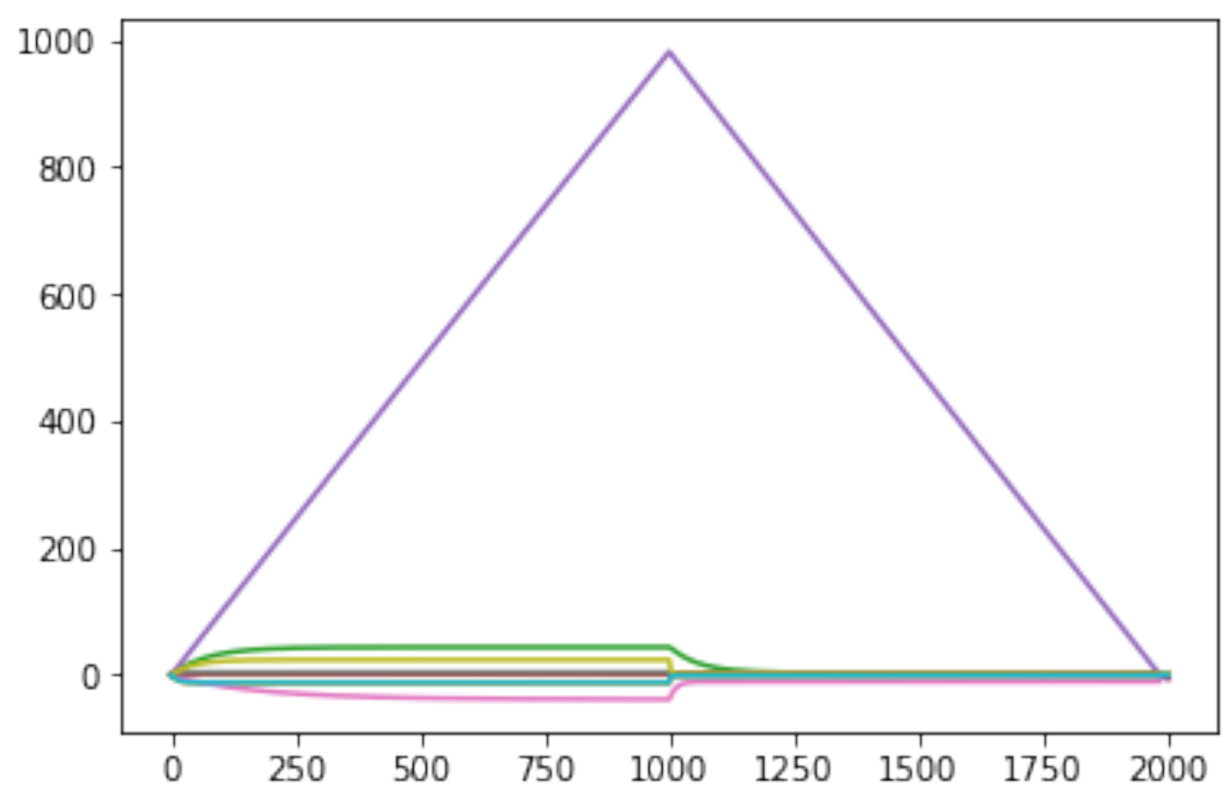
- **Took much longer to train**
- **Did not generalise even within training domain**
  - begin failing at n=39 (vs 257 for LSTM)

Empirically

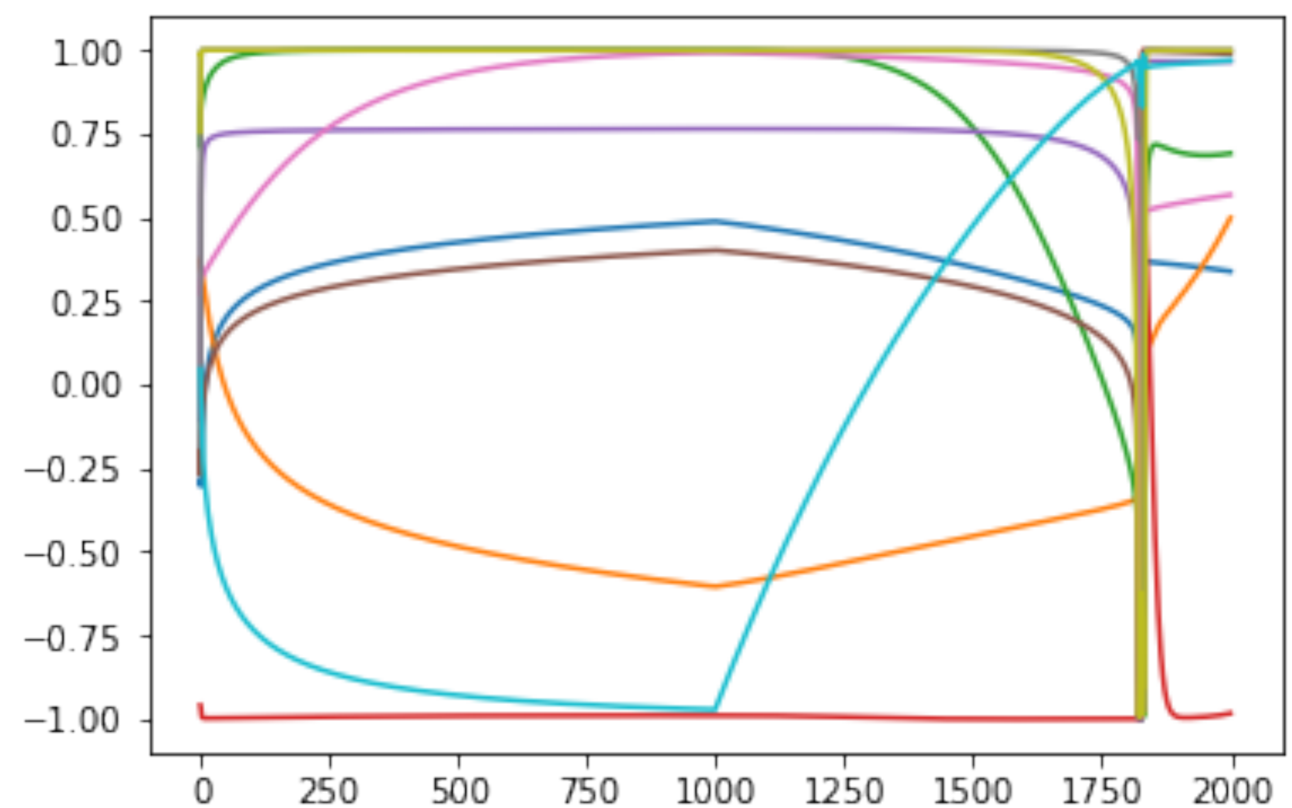
Trained $a^n b^n$**,** (on positive examples up to length 100)

Activations on $a^{1000}b^{1000}$ **:**

LSTM



GRU



GRU:

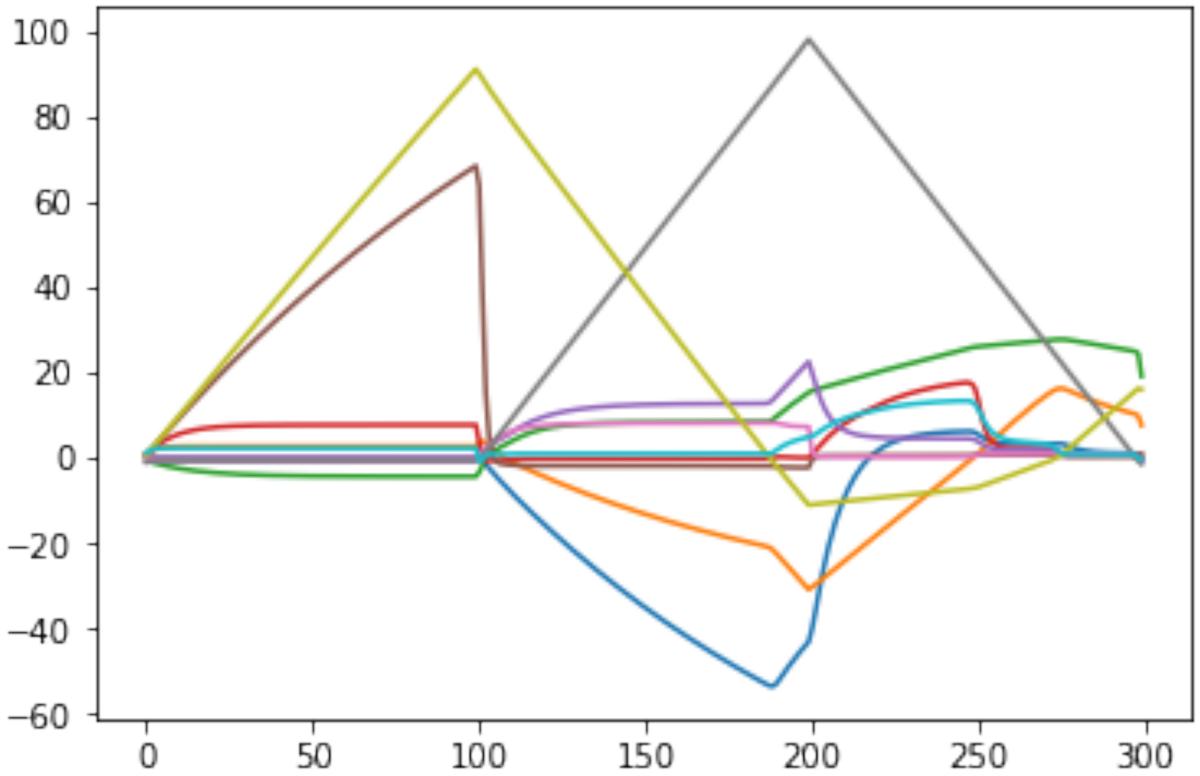
- **Took much longer to train**
- **Did not generalise even within training domain**
  - begin failing at n=39 (vs 257 for LSTM)
- **Did not learn any discernible counting mechanism**

Empirically

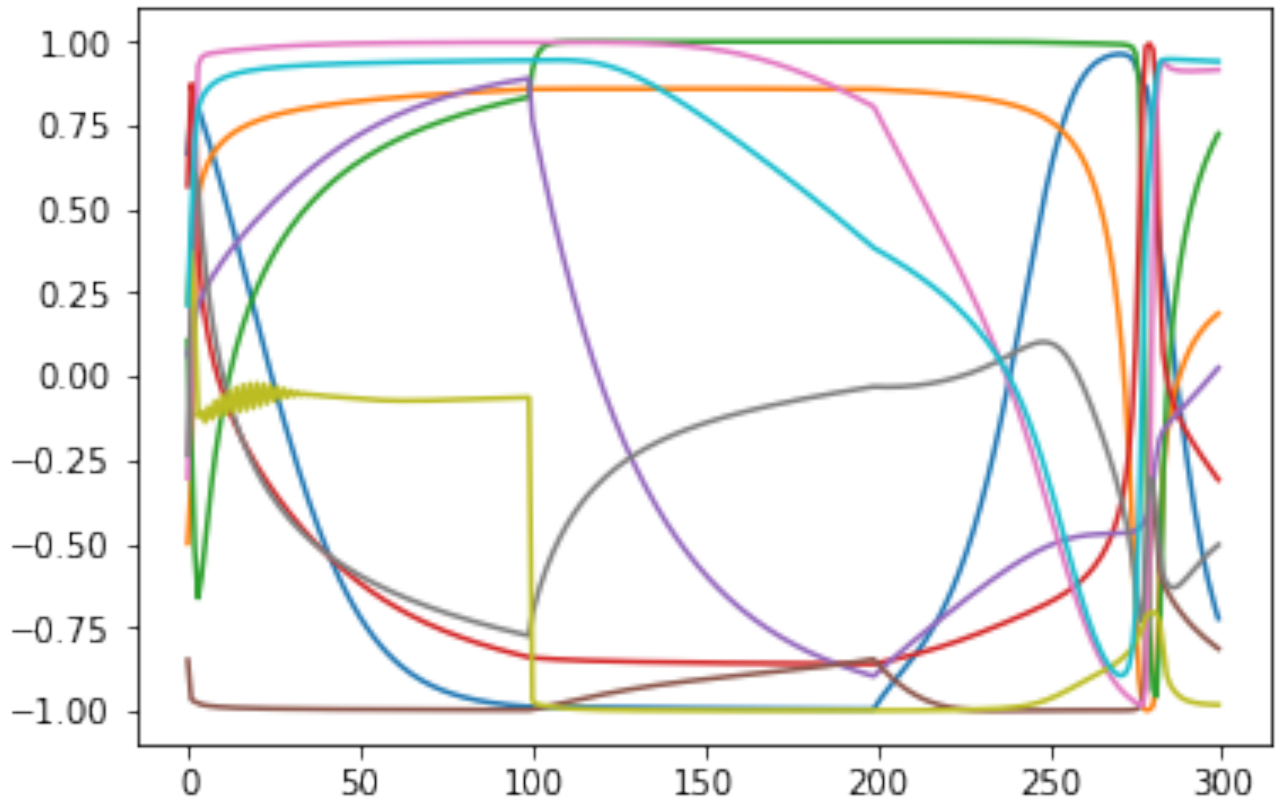
Trained $a^n b^n c^n$**,** (on positive examples up to length 50)

Activations on $a^{100} b^{100} c^{100}$**:**

LSTM



GRU

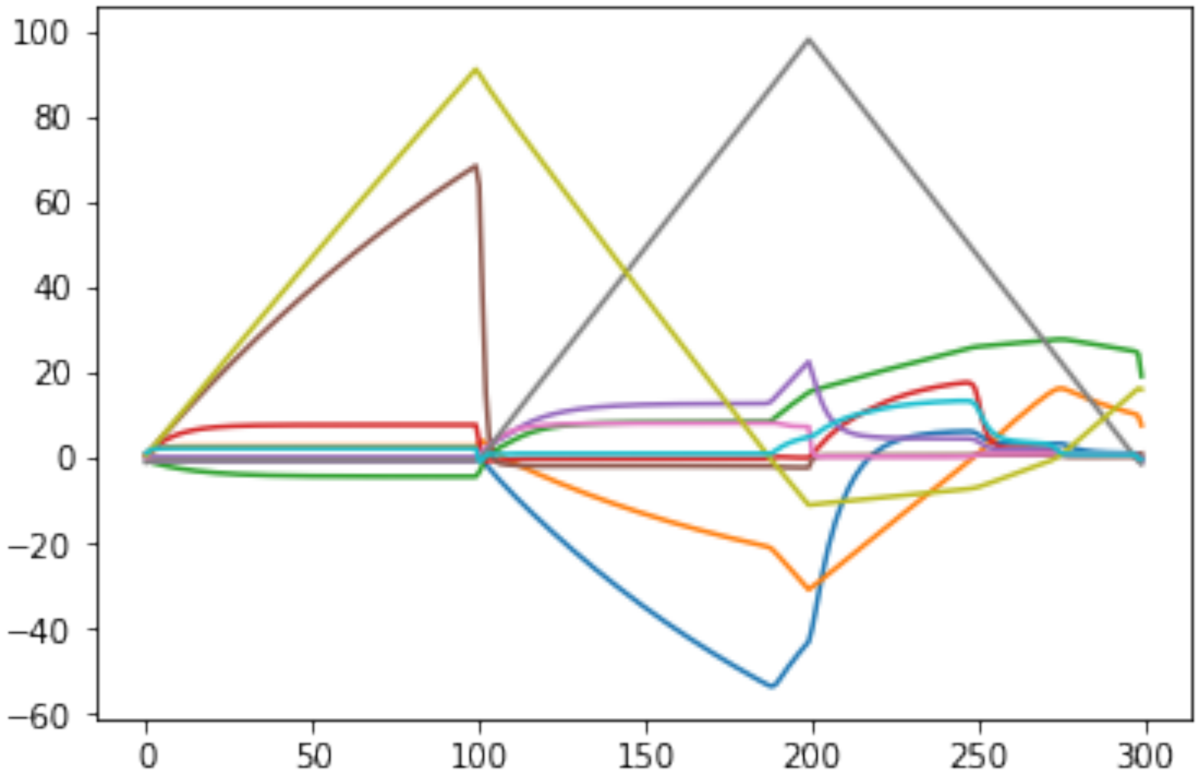


Empirically

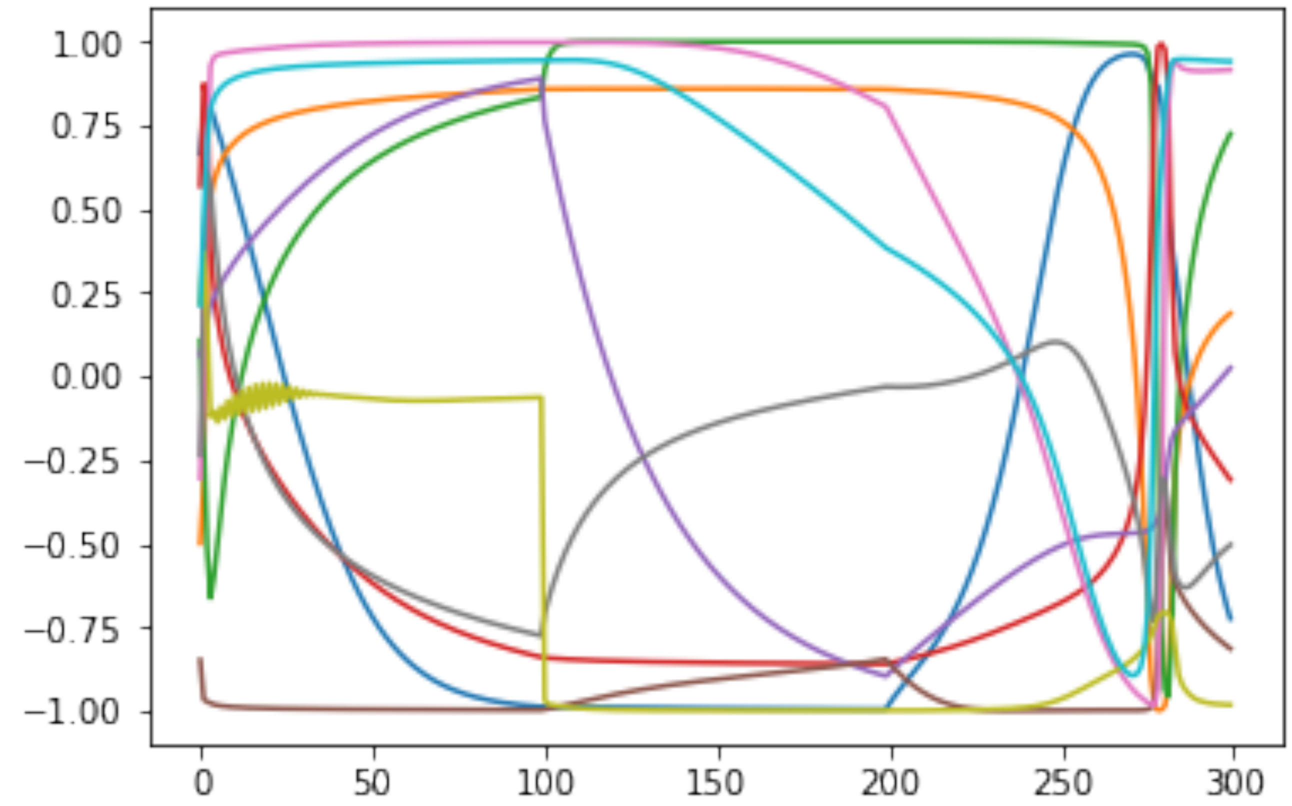
Trained $a^n b^n c^n$**,** (on positive examples up to length 100)

Activations on $a^{100} b^{100} c^{100}$**:**

LSTM



GRU

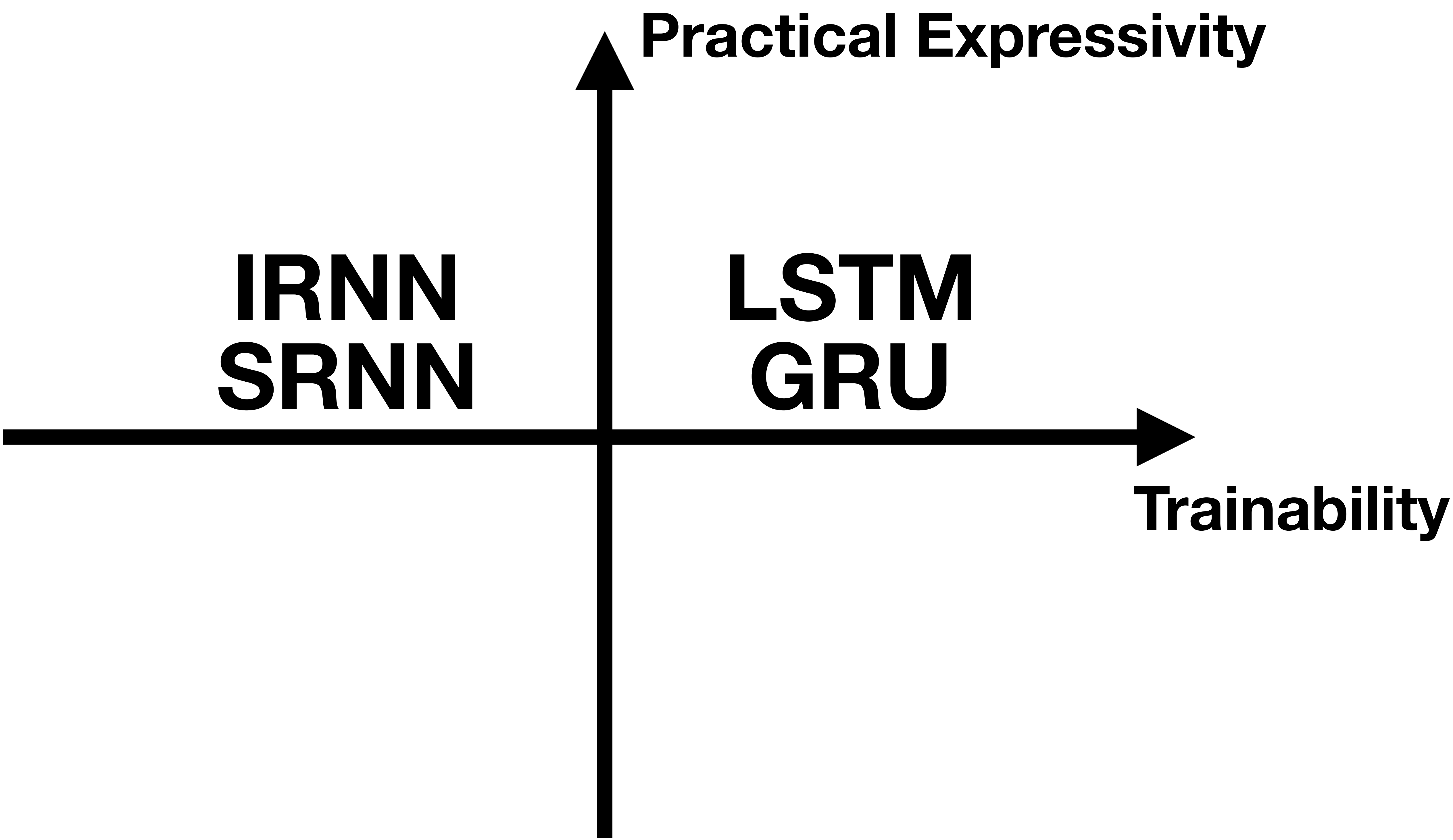


GRU:

- **Took much longer to train**
- **Did not generalise well**
  - begin failing at n=9 (vs 101 for LSTM)
- **Did not learn any discernible counting mechanism**

Conclusion

Conclusion



Take Home Message

Architectural Choices Matter!

and result in actual differences in expressive power

Don’t fall in the Turing Tarpit!

Thank You

GitHub repository:

https://github.com/tech-srl/counting_dimensions

Google Colab (link through GitHub as well):

https://tinyurl.com/ybjkumrz