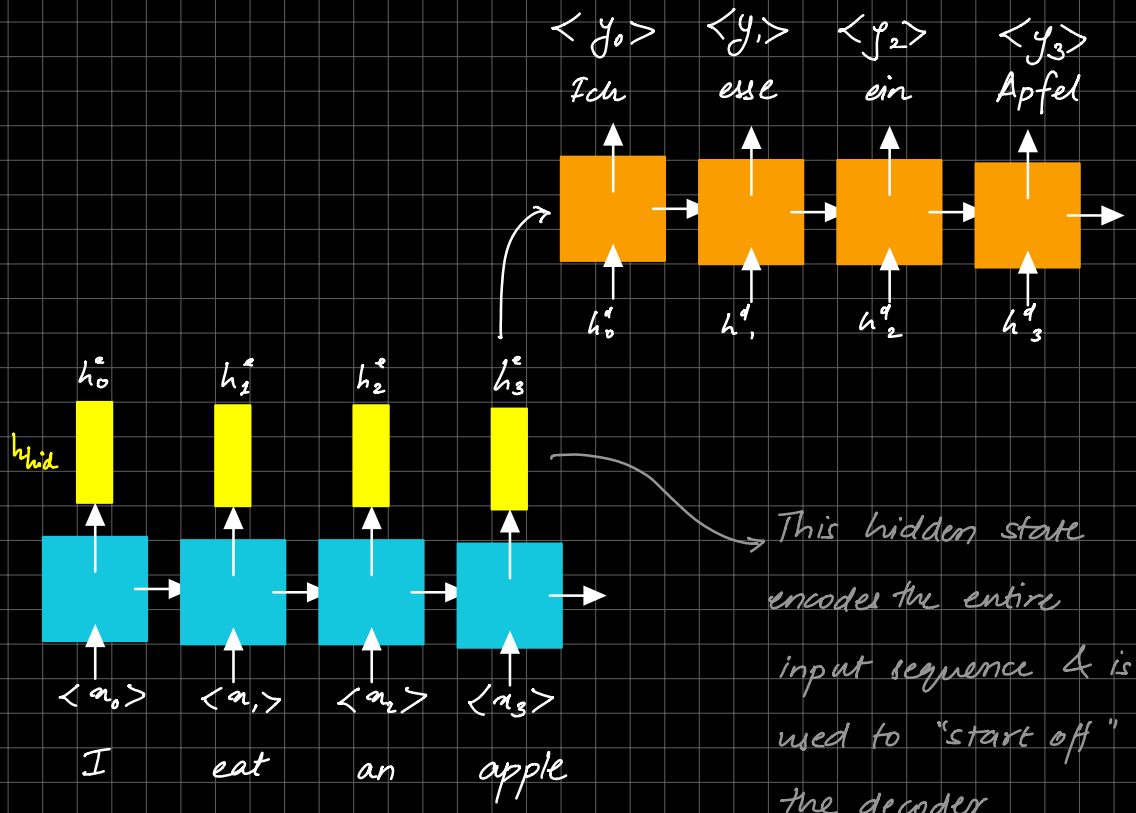
* Recitation 8 : Attention :

Encoder

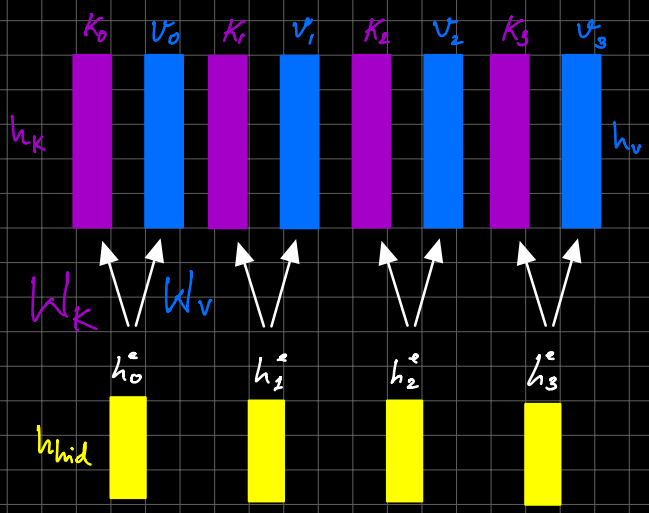
Decoder

Encoder Hidden States



This hidden state encodes the entire input sequence & is used to "start off" the decoder

"Construct Keys & Values from Encoder Hidden States"



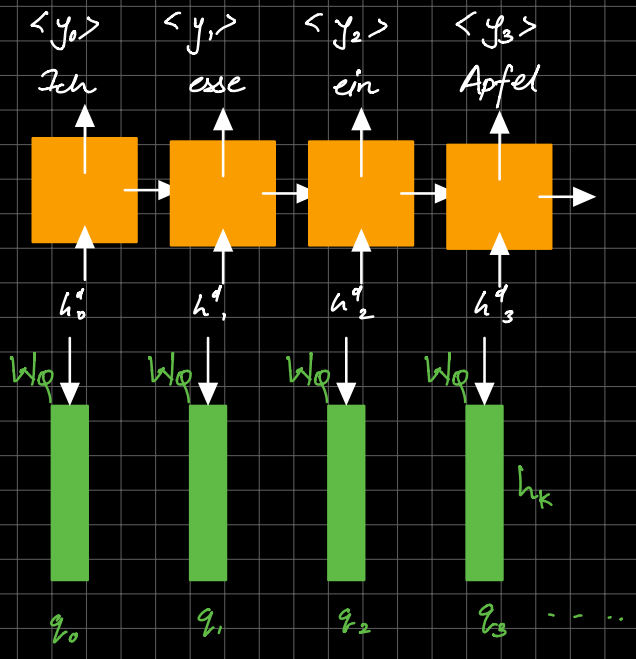
- Using matrices W_K and W_V, we construct keys & values for all the Encoder Hidden States

- Matrix Dimensions:

$W_K = h_{hid} \times h_K$

$W_V = h_{hid} \times h_V$

"Construct Queries from Decoder Hidden States"

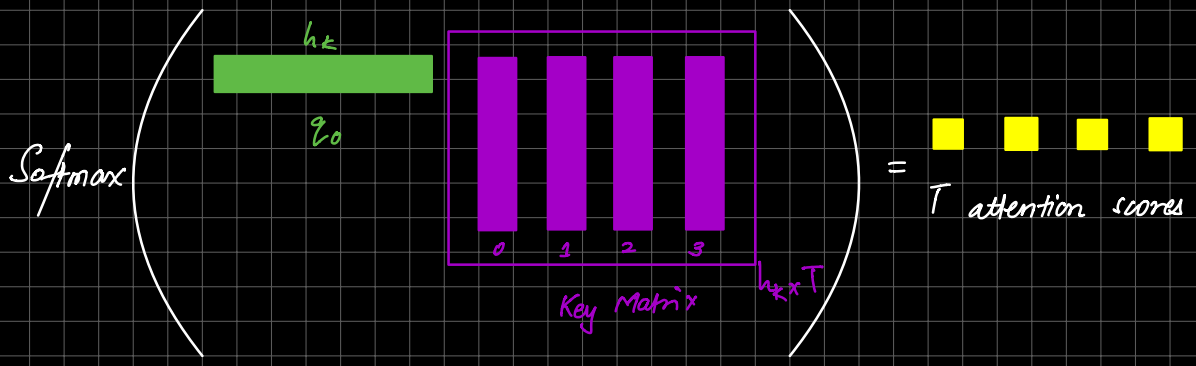


- Using matrix W_Q, we construct a query from the Decoder hidden state.

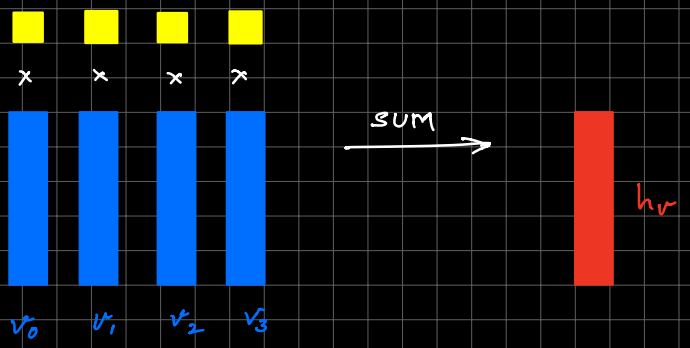
$W_Q = h_{hid} \times h_K$

- NOTE: Queries & Keys are of the same dimension in this example but this is not true for all types of attention.

"Compute Attention Scores"

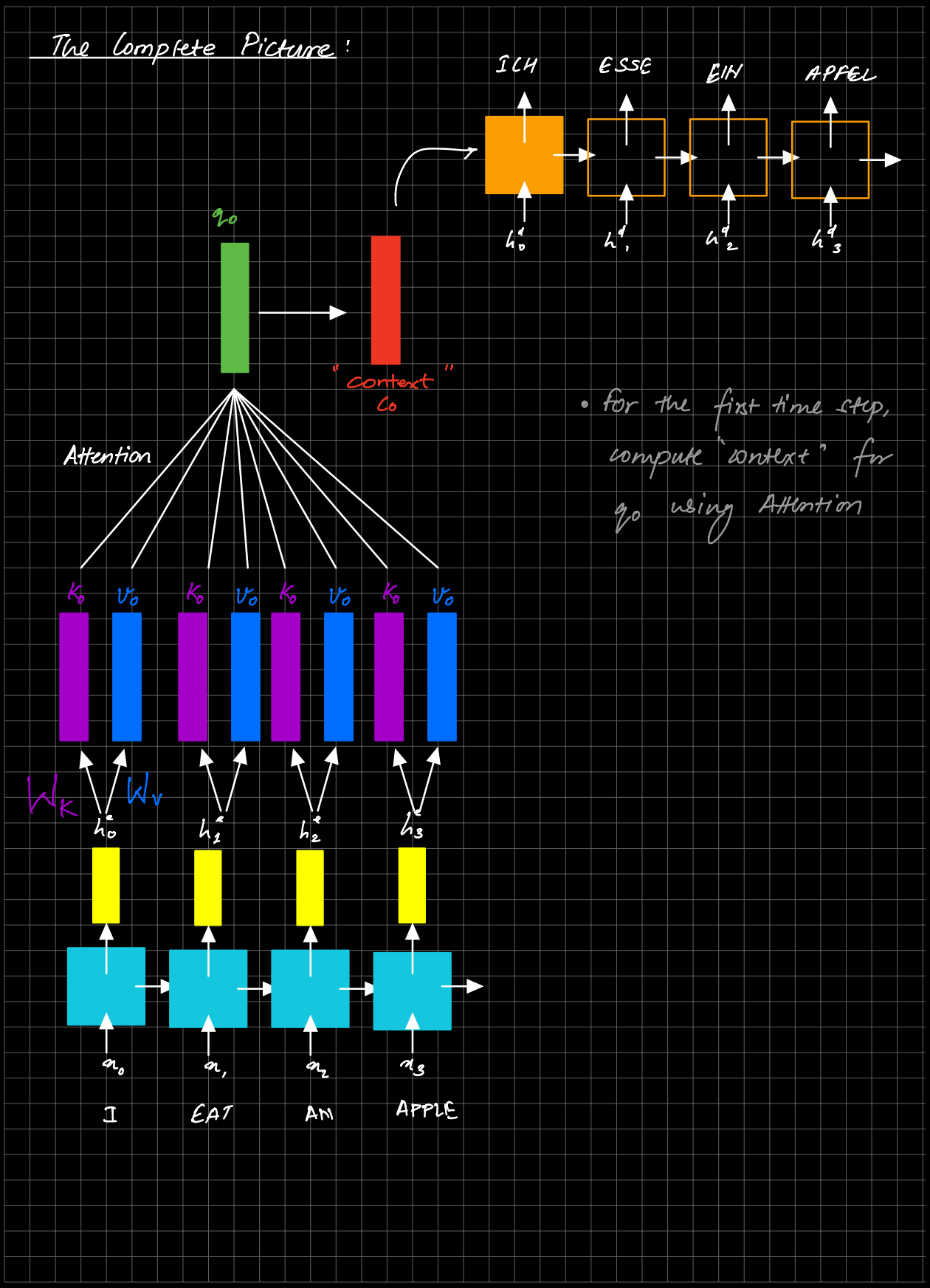


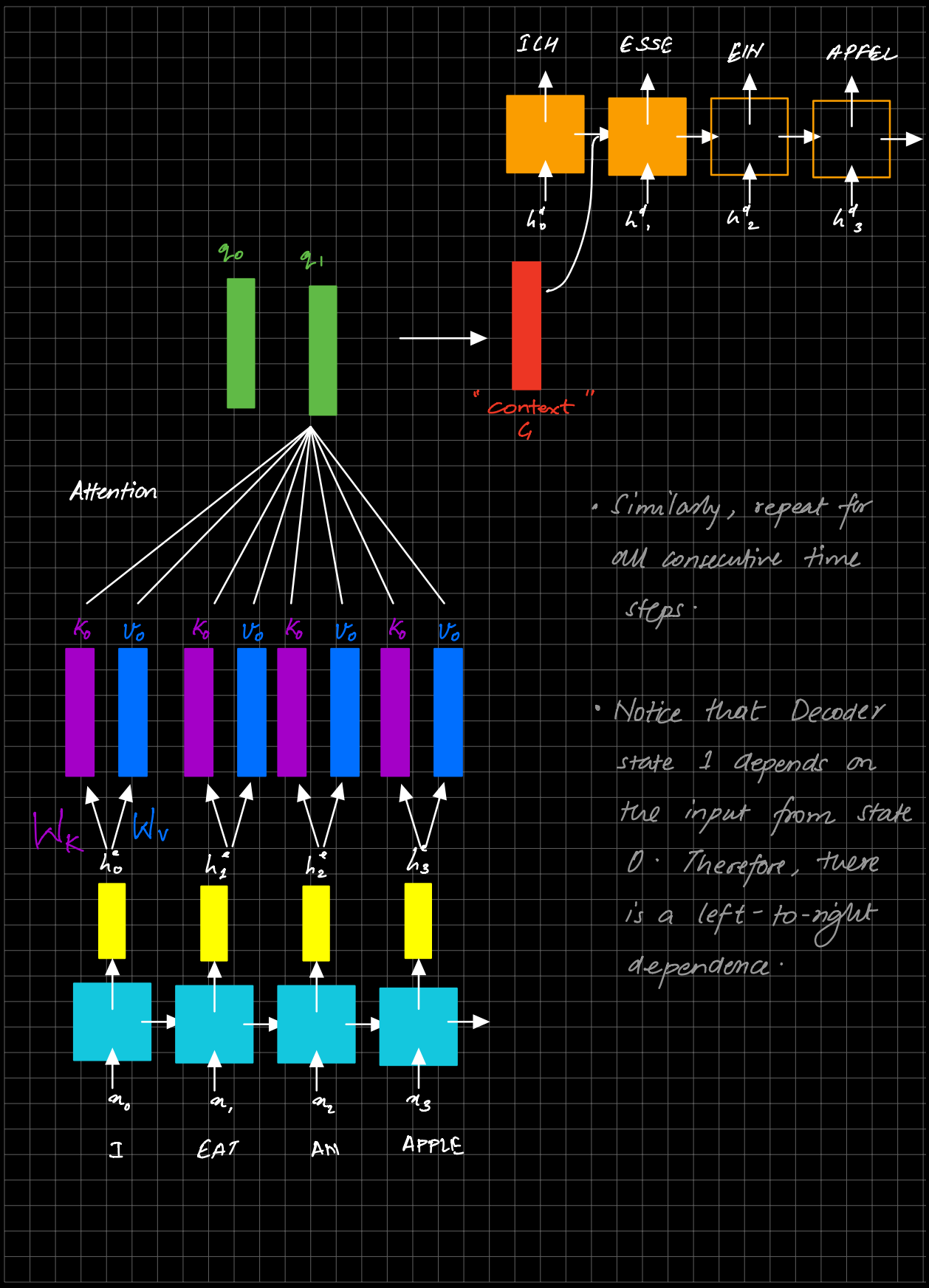
"Scale values by attention scores"



FINAL RERESENTATION

FOR ONE DECODER STATE

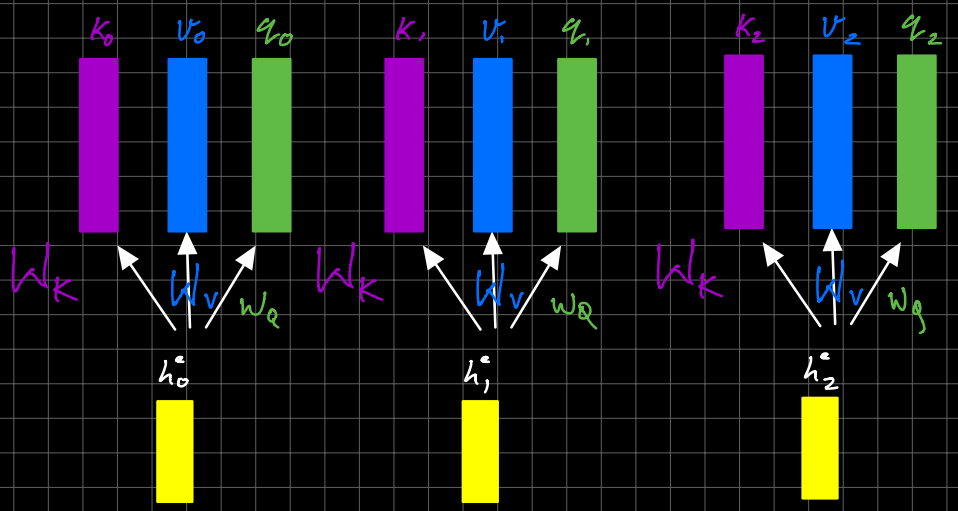

The Complete Picture!
ICH
ESSE
EIN
APFEL
h^q_0
h^q_1
h^q_2
h^q_3
q_0
"context"
c_0
Attention
k_0 v_0 k_0 v_0 k_0 v_0 k_0 v_0
W_K
W_V
h^e_0
h^e_1
h^e_2
h^e_3
a_0
a_1
a_2
a_3
I
EAT
AM
APPLE
• for the first time step, compute "context" for q_0 using Attention



- Similarly, repeat for all consecutive time steps.

- Notice that Decoder state 1 depends on the input from state 0. Therefore, there is a left-to-right dependence.

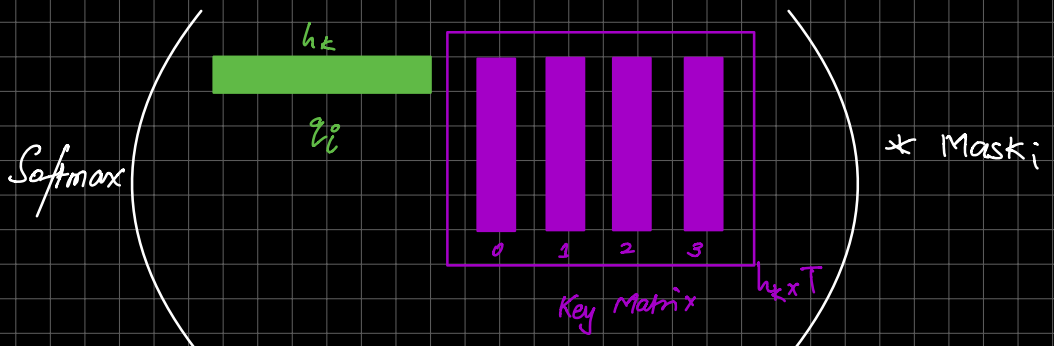
"Self Attention"



- Keys, Values, and Queries are all computed from Encoder Hidden States.
- Compute Attention between queries, keys & values exactly like before.
- BUT, no left-to-right dependence, so can compute all context vectors <u>in parallel</u>.

"Masking in Transformer Decoder"

- want to condition only on previous tokens' context. ∴ "Mask" out the keys from consecutive tokens while computing attention.



- Mask for q_0 = [1 0 0 0]
- Mask for q_1 = [1 1 0 0]
- Mask for q_2 = [1 1 1 0]
- Mask for q_3 = [1 1 1 1]

* To compute all the attention scores at once :