בס"ד

STEVE GUTFREUND – 342873791
YOEL JASNER – 204380992

1. **SUM**
The alphabet: $\sum\{0,1,2,3,4,5,6,7,8,9,\#\}$
The languages:

$$L_{good} = \{\text{a\#b\#c} | a, b, c \in \sum^{*} \setminus \{\#\}\ and\ a + b = c\}$$
$$L_{bad} = \{\text{a\#b\#c} | a, b, c \in \sum^{*} \setminus \{\#\}\ and\ a + b = c - 1\}$$

The idea of this language is that LSTM might have the ability of memory, but he will have difficulties with math equations. As there's no regular expression for this language, we believed that a LSTM will fail.

And so it did. We tried it with several parameters and sizes, we let it run for as many epoch it wants but the model stayed stuck between 48-52% accuracy. Which is the probability of guessing right the lable (1 of 2 classes).

2. **PALINDROME**
The alphabet: $\sum\{1, \dots, 9, a, \dots, z, \#\}$
The languages:

$$L_{good} = \{\text{w\#}w^{R} | w \in \sum^{*} \setminus \{\#\}\ and\ w^{R} = reversed(w)\}$$
$$L_{bad} = \{\text{w\#u} | w, u \in \sum^{*} \setminus \{\#\}\ and\ |w| = |u|\ and\ u \neq w^{R}\}$$

As our model is LSTM and not BiLSTM we believe it will be very hard to distinguish between the two languages. Since our model gets the input from left to right and has no ability to 'go back' and compare the beginning of the string with the end etc.

And so, our model had the chance to run for as many training-iterations as it needed, on as many training examples it wanted, it didn't get higher than 53% accuracy, which as before is equal to the probability of choosing randomly the right class.

3. **MUL**
The alphabet: $\sum\{0,1,2,3,4,5,6,7,8,9,\#\}$
The languages:

$$L_{good} = \{a\#b\#c \,\big|\, a, b, c \in \sum^{*} \setminus \{\#\}\ \ and\ a * b = c\}$$
$$L_{bad} = \{\text{a\#b\#c} | a, b, c \in \sum^{*} \setminus \{\#\}\ and\ a * b = c - 1\}$$

The idea is quite the same as SUM just with multiplication. We believe a LSTM model cannot solve math problems.

To our surprise, the model performed better than on SUM. The model achieved around 76% accuracy. After about 10 epochs, the loss on the dev set started to go up and the accuracy down, a sign of over-fitting.
Changing parameters or sizes did not help in any way.
Anyway, 76% is not enough, so we successfully failed the model again.