# Mathematical Formulations for Fine-Tuning Vulnerabilities Dataset

## Tokenization and Input Representation

Given a code snippet $C$, the tokenizer splits it into tokens:

$$T = \{t_1, t_2, \ldots, t_k\}$$

where $k \leq 512$. If $k < 512$, padding tokens `<PAD>` are appended:

$$T_{\text{padded}} = \{t_1, t_2, \ldots, t_k, \underbrace{\texttt{<PAD>}, \ldots, \texttt{<PAD>}}_{512-k}\}$$

## Model Architecture

The model processes the tokenized input and computes logits for classification:

$$\text{logits} = W \cdot h + b$$

where:

- $h$: Hidden state from the final transformer layer.
- $W, b$: Weights and biases of the classification head.

The logits are transformed into probabilities using the softmax function:

$$P(y = i|x) = \frac{e^{\text{logit}_i}}{\sum_j e^{\text{logit}_j}}$$

## Loss Function

The Cross-Entropy Loss is used to optimize the model:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c})$$

where:

- $N$: Number of samples.
- $C$: Number of classes.
- $y_{i,c}$: True label (one-hot encoded).
- $\hat{y}_{i,c}$: Predicted probability for class $c$.

# Evaluation Metrics

## Accuracy

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

## Precision

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

## Recall

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

## F1-Score

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$