**N-Order Adjacency-Frequency Data-Structure Proof**

**Andrew Joseph Heroux**

**Introduction:**

The purpose of the Adjacency-Frequency Data-Structure (Adjacency-Struct) is to be able to look back at the last n tokens (measure/interval) in a given song and select a token that is likely to be adjacent to a sequence of n tokens in a given data-set. This is achieved using JSON (Java Script Object Notation.) Every JSON object is structured with a signature, which is the sequence of tokens, and two parallel arrays. One array stores the signature of adjacent tokens and the other stores the frequencies of the signatures in the first array.

**1-Order Example:**

Example Data: The July sun caused a fragment of black pine wax to ooze on the velvet quilt.

|  |  |
| --- | --- |
| **Signature ->** | the |
| **Adjacent Signatures \/** | **Signature Frequencies \/** |
| july | 1 |
| velvet | 1 |

In this example of a 1-order system the signature is “the” and the adjacent frequencies, “july” and “velvet,” each have frequencies of one because they only appear once after “the.” If the data was “The July the July the July,” the frequency for “july” would be three.

**Order-3/N-Order Example:**

In an order-3 system the signature is a sequence of three tokens. Here is an example of an order-3 system.

Example Data: Go to the hardware store to look for five-volt resistors. If you cannot find them there, then go to the computer repair shop.

|  |  |
| --- | --- |
| **Signature ->** | go to the |
| **Adjacent Signatures \/** | **Signature Frequencies \/** |
| hardware | 1 |
| computer | 1 |

To expand from an order-3 to an n-order system, the program only needs to change the number of tokens in the sequence of signatures.

**1-Order System Proof:**