# The Unicon JSON Validator

By Hunter Thomas, John Haag, and Zachary Kovalenko

## Unicon Technical Report

May 11, 2021

### Abstract

This report describes a package called The Unicon JSON (JavaScript Object Notation) Schema Validator. This package intakes a JSON Object file and a JSON Schema file compare’s the two and outputs to the user whether or not the Object file conforms to the Schema file.

SUNY College at Brockport

Department of Computer Science

Brockport, NY 14420, USA

## Introduction

JSON is a lightweight data-interchange format that is both human and machine readable. This report describes a Unicon function that validates a JSON Object file from a JSON Schema file.

## An Overview of the Unicon JSON Validator

The Unicon JSON Validator is a single function package. The function requires two JSON files for proper execution, the first is a JSON Object file, and the second a JSON Schema file. These two JSON files are compared line by line, based on JSON validation keywords, and certain attributes to ascertain whether or not the Object file is indeed conforming to the Schema file. The validation is returned to the user in the form of a string, and either “Validation Passed” or “Validation Failed” are printed to the command line. This function intakes two command line arguments, “ -f ” for the JSON Object file name, and “ -s ” for the JSON Schema file name. In order to run this function, the user must have the JSON Validator, and both the Object and Schema files in the same directory as the Unicon IDE. First the user must compile the program. Then the user may run the program with the above stated command line arguments followed by the name of each specified file. The following example shows how to run this function on the Linux, and Apple operating systems in the command line:

unicon jsonValidation.icn

./jsonValidation -f array.json -s array.schema.json

The following example shows how to run this function on a Windows operating system:

unicon jsonValidation.icn

jsonValidation.exe -f array.json -s array.schema.json