Notation 3 (N3) rule language:

- Standard N3 triple format:

  SUBJECT PROPERTY OBJECT

- Using prefix's:
  Prefixes can be used to abbreviate url's used in a given N3 file. For example:

  Instead of using:

```
<example#Bob>
      <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
            <example#Person>.
```

  The following prefixes can be implemented to shorten the statement:

```
@prefix : <example#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

:Bob rdf:type :Person.
```

- Representing variables:
  When using rules, variables can be used to test subjects against them. In N3, a variable is represented with a ? followed by characters used to represent it. For example:

```
@prefix : <example#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

{?S rdf:type :Swan} => {?S :color :White}.
```

  This means that all subjects who have a rdf:type of :Swan, then they also will have the :color property of object:White.

Triples written for Euler generally follow the format of axioms and queries, where the axioms contain relations and facts regarding the knowledge base, and the query is to retrieve the triples from it.

Axioms:

A standard axiom contains a “facts” and a “relations” file.

- Relations:
  The relations file lists all the relationships between the triples in the knowledge base (using inference). It can also be used to define any initial properties associated to subjects. All relations and initial values are written in the form of rules and have this form:

  ```
  {SUBJECT PROPERTY OBJECT} => {SUBJECT PROPERTY OBJECT}.
  ```

  For example, this relation will imply that if a certain individual C, has a father F, then F must have a child C:

  ```
  @prefix : <example#>.
  @prefix log: <http://www.w3.org/2000/10/swap/log#>.

  {?C :father ?F} log:implies {?F :child ?C}
  ```

  The ? symbol indicates a variable used in the formula. Therefore, ?C means every subject in the facts file is checked to see if they have a property of :father and if it has a value.

  log:implies is equivalent to => in the N3 language. Therefore, => can be used in rules to represent an implication.

- Facts:
  A facts file could be thought of as creating an instance of a class such as in object-oriented programming. New subjects are created, which are generally given an rdf:type and other property values. These will be used with the query later to draw conclusions about these facts.