Documentation

Introduction

Racer is a language that lets you describe ontologies and the interactions between them. It resembles SQL in a way that the queries look alike.

Since it is a description language it comes with an IDE that lets you develop your own ontologies and also with CLI if you want to write code in your favorite text editor. Unfortunately even if you can write your code in a text editor it does not come with a linter or syntax verifier.

Here comes into play this project. It is a visual studio code extension that will help the developer to learn the language faster with its intuitive syntax checker as well as the code coloring.

Technical Depth

The technical part was made available with the help of the code server provided by microsoft. To process the input file the AST structure was used.

The AST (Abstract syntax tree) has as nodes different tokens that are used to define a structure. Let's define a Tree constructor such as this V X belongs to a syntactically correct tree then construct(X) = Y and assemble(Y) = X. where construct function can return only correct AST's and assemble takes a correctly defined AST and returns the initial string that was deconstructed.

Here is an example.
Let x be defined as below.

```
(full-reset)
( implies A B)
(concept-instances R)
```

And an AST schema looks like this.
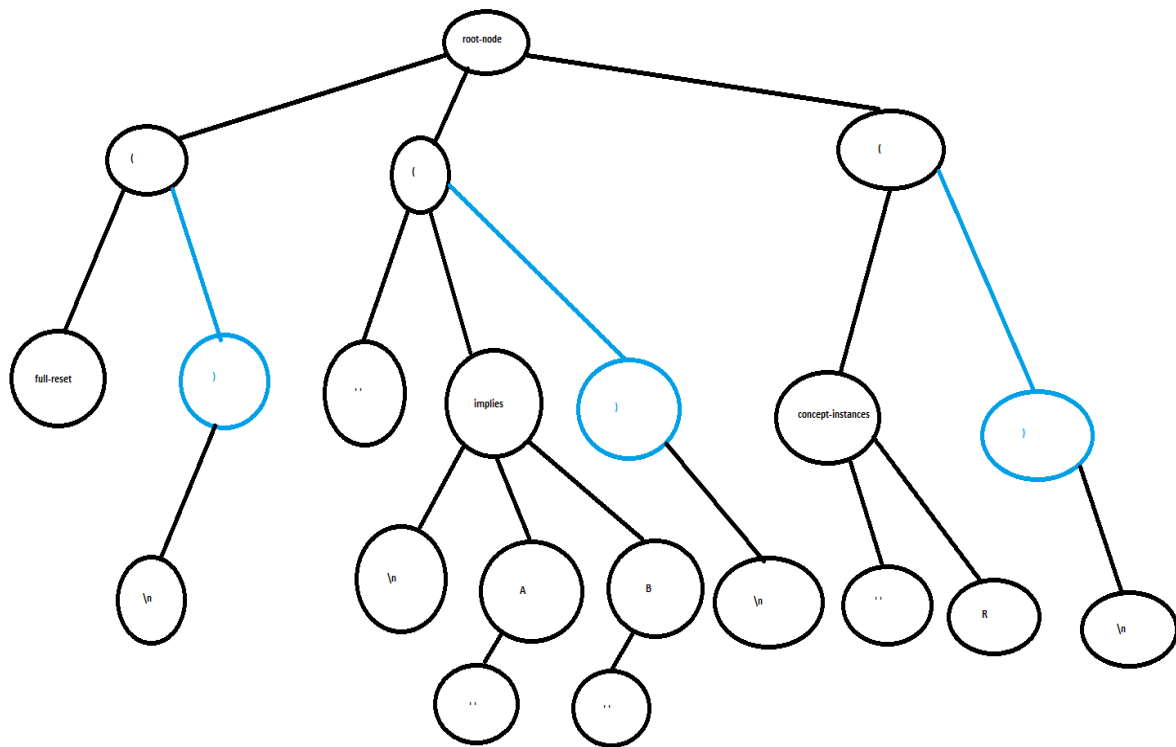
The blue illustrations indicate a complementary token.
In this illustration each node indicates a node and their children, for example the
implies token has the empty space between itself and the **A** parameter. The **A**
parameter has a children parameter, the empty space between itself and the **B**
parameter.
The parser files reside in the **src/** folder and the function that creates this tree is called
parser_CreateAST.

Conclusions

Even tho this is just a demo, I believe that, as a concept, this can become a
powerful tool in the future, enabling more developers to tap in this technology.
It will also help the future developers to easily learn this new technology since it will not
let them write syntactically incorrect code.