Antephanie

By Anfal Al-Hussaini and Stephanie Motz

Steps to Building the Parser

- **Find a Parser Generator**
- **Define Grammar**: Started by defining the grammar of 'Antephanie'
- **Choose Parsing Technique**: Resursive Top-Down Approach (starts at the beginning and follows the rules step by step until it understands the whole language)
- **Implement Lexer**
- **Implement Parser**: Analyze token stream from lexer and construct a parse tree or AST representing the structure of the input.
- **Handle Errors**: include providing informative error messages and suggestions for corrections.
- **Test**: Write test cases to verify that the parser behaves correctly. Test both the Lexer and Parser components thoroughly.
- **Documentation**

Updated Token Types

Java Decision-Making	Antephanie Decision-Making
IF	Reactant
ELSE	Product
SWITCH	Experiment
CASE	Reaction
BREAK	Spill

Java Assignment	Antephanie Assignment
=	=
*=	[Ts=]
/=	[Dy=]
+=	[P=]
-=	[-=]

Java Logical	Antephanie Logical
OR	[O]
AND	[Am]
NOT	[No]

Java Variable	Antephanie Variable
Float	Independent

Looping Java	Looping Antephanie
For	Yields
While	While
Do-While	Conduct-While

Sample Code- Parser

Example of if-then statement:

```
independent weight = 1.0078;
%Takes the FLOAT variable weight
and assigns it 1.0078.

reactants (weight <= 1.0078) {
        formula("hydrogen");
    }
    products {
        formula("not hydrogen");
    }
```

Example of Switch Case:

```
experiments (groups) {

    reaction 1:
       element = "metals";
       spill;
    reaction 2:
        element = "nonmetals";
        spill;
    reaction 3:
        element = "noble gases";
        spill;
    }
```

Example of While and Do-While:

```
independent weight =
107.8682;
while(weight < 108.0000) {
    formula(weight);
      weight[In]
}

conduct {
    formula("silver");
} while (element = "Ag");
```

Demo