1 Language Description

People are sometimes suffering from balancing chemical equations, especially when it comes to complicated ones (i.e. $10\text{Al} + 6\text{NaNO}_3 + 6\text{H}_2\text{O} = 4\text{Al(OH)}_3 + 3\text{N}_2 + 6\text{NaAlO}_2$). In this project, we try to design a programming language that aims at this kind of problems, along with some other chemical calculations.

Our language, "iChemi", will provide programmers with a built-in function that will serve to balance equations and with data structures that can store information collected from users and programmers effectively. Flow controls will also be provided so that the program can decide, based on reality, what products will be produced under different material usage ratios, reaction conditions, etc.

Due to the given features, the language will be very concise and easy to use for programmers.

2 Example Code

The goal of iChemi is to describe chemical equations in a programming way and provide users with simple language constructs to manipulate chemical related formulas.

Describing Equation

Here is a simple example that shows how to describe a balanced equation using iChemi language. The chemical equation we will describe is $2\text{H}_2 + \text{O}_2 = 2\text{H}_2\text{O}$.

```plaintext
1 Equation E1{
2   Left:   [(2)H^2, O^2];
3   Right: [(2)H^2_O];
4 }
```

$E1$ is the structure of a chemical equation. The elements in the $Left$ attribute are reactants and those in the $Right$ attribute are products. Each element consists of two parts, one is coefficient (could be omitted when it is 1) and the other is molecular formula. The coefficient
of a certain molecular formula is placed in front of the molecular formula within a pair of brackets. Different chemical elements in a certain molecular formula should be separated by a ‘_’. If a molecule contains more than one atom of a particular element, this quantity is indicated using a number following a ‘^’ symbol.

**Balancing Equation**
This example shows how to balance an equation using iChemi’s function.

Given an equation \( \text{NaOH} + \text{CO}_2 \rightarrow \text{Na}_2\text{CO}_3 + \text{H}_2\text{O} \), we describe it as:

```plaintext
1 Equation E2{
2   Left: [Na_O_H, C_O^2];
3   Right: [Na^2_C_O^3, H^2_O];
4 }
```

After declaration, we balance it by calling \( E2.balance() \). Then using function \( write(E2) \), we could get the answer: \( 2\text{NaOH} + \text{CO}_2 = \text{Na}_2\text{CO}_3 + \text{H}_2\text{O} \).

**Using Conditional Statement**
It is known that the reaction between \( \text{NaOH} \) and \( \text{CO}_2 \) will produce different products (\( \text{Na}_2\text{CO}_3 \) or \( \text{NaHCO}_3 \) or both) under different ratios of reactants. iChemi provides users with flow control expression like \( if \cdots else \cdots \) to describe these different equations.

```plaintext
1 # In this case, we only know the reactants.
2
3 Equation E3{
4   Left: [Na_O_H, C_O^2];
5 }

6 # Get the amount of reactants from user's input.
7
8 main() {
9   write("Input the amount of NaOH(mol)"咥);
10  int a = read();
11  write("Input the amount of CO2(mol)"咥);
12  int b = read();
13
14  # Analyze the products according to the inputs.
15  if(a/b <=1){
16      E3. Right.add(Na_H_C_O^3);
17  }
18
19  else if(a/b>=2){
20      E3. Right.add(Na^2_C_O^3);
21  }
22
23  else{
24      E3. Right.add(Na_H_C_O^3);
25      E3. Right.add(Na^2_C_O^3);
26  }
27  # balance function might return 1 of these 3 values: 0, 1 and -1.
28  int results = E. balance();
29  #equation could be balanced
```

2
if (results == 0) {
    write(E3);
}  

#equation could not be balanced
else if (results == -1){
    write("Cannot balance the equation!");
}  

#more than one balance solutions to the equation
else{
    write("More than one solutions to the equation!");
    #write any one of the solutions.
    write(E3);
}  

3  iChemi Syntax

Comment:  # We will use single line comment, prefaced by the # symbol.

Keyword:  Equation, Left, Right, main, int, float, if, else, read, write.

Operator:  + - * / > < = <= >= != == || && , “ ”

Declaration:

a. Variable declaration:
1  [basic_data_type] variable_name;
2  [basic_data_type] variable_name = value;

b. Chemical equation declaration:
1  Equation A{
2      Left: [element, element];
3      #Right(product) is not required.
4      Right: [element, element];
5  }

Note: element is consist of coefficient and molecular formula (atoms and subscripts).

Statement:  A statement ends with a ; symbol.

Control Flow:
1  if (boolean){
2      statement;
3  }
4  else{
5      statement;
6  }
if (boolean) {
    statement;
} else if (boolean) {
    statement;
} else {
    statement;
}

**Class:** Equation.
**Attributes:** Left, Right;
**Function:** Balance().
The values returned by function balance could be 0, 1, -1.
0: Indicates that there is exactly one solution to the equation.
1: Indicates that there are many solutions to the equation.
-1: Indicates that there is no solution to the equation.

**Build-in Function:** read(), write().