

苏州大学实验报告

院系	计算机学院	年级专业	21 计科	姓名	方浩楠	学号	2127405048
课程名称	编译原理课程实践					成绩	
指导教师	王中卿	同组实验者	无	实验日期	2023. 10. 23		

实验名称 实验 7 化学分子式解析

一. 实验目的

- 1.学习和理解编译原理中的词法分析和语法分析的基本概念和实现方法。
- 2.掌握使用 Python Lex-Yacc (PLY) 库进行编译器前端设计的基本技能。
- 3.通过实际编程练习,了解如何构建一个简单的化学分子式解析器,并计算分子式中的元素数量。

二. 实验内容

- 1.熟悉 PLY 库的基本编写规范,阅读 PLY 使用手册,了解 PLY 的词法分析和语法分析的基本使用方法。
- 2.分析四则运算的程序,并基于此设计和实现一个能够解析化学分子式,并计算其中元素数量的程序。
- 3.设计测试用例,验证解析器的正确性和可靠性。

三. 实验步骤和结果

本项目的所有 python 代码均根据 **Google Style Guide** 给出了 docstring 和注释,便于阅读和理解。
同时给出了 **readme.md** 用来解释项目,并且给出项目的运行方式,同时便于快速理解项目的结构和功能

项目结构:

编译原理实验 7 ChemicalFormulaParser/

```
├── src/
│   ├── lexer.py      # 词法分析器模块
│   ├── parser.py     # 语法分析器模块
│   └── main.py       # 主执行模块, 包含 atom_count 函数
├── tests/
│   └── test_parser.py # 测试代码,用来测试给出的样例
├── docs/
│   ├── 实验要求.pdf   # 本次编译原理实验的实验要求
│   ├── 实验 7 实验报告.pdf # 本次编译原理实验的实验报告的 pdf 格式
│   └── 实验 7 实验报告.docx # 本次编译原理实验的实验报告 docx 格式
├── run.py            # 顶级脚本, 用于运行项目
└── readme.md.       # readme 文件,用来项目概述和使用说明
```

src/文件夹下存放着项目文件,包括词法分析和语法分析的模块

tests/文件夹下存放着测试代码
run.py 用来运行该项目
readme.md 用来对该项目进行概述,同时给出使用说明

项目运行方式:
在项目根目录运行
python3 run.py

项目测试方式:
在项目根目录运行

python3 -m pytest

1.环境准备和依赖安装:
安装 PLY 库以支持词法和语法分析:

pip3 install ply

安装 **pytest** 以便于测试

pip3 install pytest

2. 设计和实现词法分析器 (./src/lexer.py)

模块目的:

该模块负责识别化学分子式中的元素符号和数字。
定义元素符号和数字的正则表达式匹配规则。

重要函数:

t_SYMBOL(t): 识别元素符号, 例如"C", "H", "O"等。
t_COUNT(t): 识别元素数量, 例如"2", "3", "4"等。

3.设计和实现语法分析器 (./src/parser.py)

模块目的:

该模块负责根据词法分析器的输出解析化学分子式的结构。
计算化学分子式中的元素数量。

重要函数:

p_species_list(p): 解析元素列表, 处理多个元素的情况。
p_species(p): 解析单个元素, 处理元素符号和数量。

主函数和接口设计 (src/main.py)

模块目的:

提供解析器的主要接口函数 **atom_count**, 使用户能够通过简单的函数调用来解析化学分子式和计算元素数量。

重要函数:

atom_count(formula): 解析给定的化学分子式，并计算其中的元素数量。

atom_count(formula)的实现方式:

```
def atom_count(formula):  
    """  
    计算给定化学分子式中的元素数量。  
  
    Args:  
    formula (str): 待分析的化学分子式。  
  
    Returns:  
    int: 元素的数量。  
    """  
    lexer.input(formula)  
    return parser.parse(lexer=lexer)
```

其中,parser 模块的 docstring 如下:

```
"""  
parser.py: 包含化学分子式解析器的语法分析器组件。
```

此模块定义了一个语法分析器，用于解析化学分子式并计算其中的原子数量。

Classes:

Atom: 表示化学分子式中的一个原子，包括元素符号和数量。

Functions:

p_species_list_multiple(p): 处理多个化学物种的情况。

p_species_list_single(p): 处理单个化学物种的情况。

p_species_symbol(p): 处理单个符号的情况。

p_species_symbol_count(p): 处理符号和数量的情况。

p_error(p): 语法错误处理函数。

```
"""
```

编写和运行测试代码 (tests/test_parser.py 和 run.py):

在./tests/test_parser.py 中编写单元测试，验证 atom_count 函数的正确性。

通过运行./run.py 来执行项目，并观察输出结果，验证解析器的正确性。

测试样例均是根据实验要求而给出的,并且全部通过

测试代码:

```
"""
```

test_parser.py: 包含对 main 模块中 atom_count 函数的测试用例。

此模块包含了对 atom_count 函数的多个单元测试,确保 atom_count 函数在各种输入条件下都能正确工作。

Usage:

```
python3 -m pytest # 从项目根目录运行所有测试
```

```
"""
```

```
import pytest
from src.main import atom_count
```

```
@pytest.mark.parametrize(
    "formula, expected", [
        ("He", 1),
        ("H2", 2),
        ("H2SO4", 7),
        ("CH3COOH", 8),
        ("NaCl", 2),
        ("C60H60", 120),
    ]
)
def test_molecule(formula, expected):
    assert atom_count(formula) == expected
```

测试结果:

```
===== test session starts =====
collecting ... collected 6 items
```

```
test_parser.py::test_molecule[He-1] PASSED [ 16%]
test_parser.py::test_molecule[H2-2] PASSED [ 33%]
test_parser.py::test_molecule[H2SO4-7] PASSED [ 50%]
test_parser.py::test_molecule[CH3COOH-8] PASSED [ 66%]
test_parser.py::test_molecule[NaCl-2] PASSED [ 83%]
test_parser.py::test_molecule[C60H60-120] PASSED [100%]
```

```
===== 6 passed in 0.01s =====
进程已结束,退出代码为 0
```

```

Testing started at 12:19 ...
Launching pytest with arguments test_parser.py::test_molecule --no-header --no-summary -q in /Users/fang

===== test session starts =====
collecting ... collected 6 items

test_parser.py::test_molecule[He-1] PASSED [ 16%]
test_parser.py::test_molecule[H2-2] PASSED [ 33%]
test_parser.py::test_molecule[H2SO4-7] PASSED [ 50%]
test_parser.py::test_molecule[CH3COOH-8] PASSED [ 66%]
test_parser.py::test_molecule[NaCl-2] PASSED [ 83%]
test_parser.py::test_molecule[C60H60-120] PASSED [100%]

===== 6 passed in 0.01s =====

进程已结束，退出代码为 0

```

测试样例全部通过

运行代码:./run.py

run.py 内容:

"""

run.py: 包含执行化学分子式解析器的主入口。

此脚本为化学分子式解析器提供了一个简单的用户界面。

用户可以输入一个化学分子式，脚本将返回该分子式中的原子数量。

Usage:

python run.py # 从命令行运行此脚本，按提示输入化学分子式

"""

from src.main import atom_count

def main() -> None:

"""

主函数，提示用户输入化学分子式，并输出原子数量。

从用户获取化学分子式的输入，调用 atom_count 函数来计算原子数量，然后将结果打印到控制台。

Args:

None

Returns:

None.但是会在终端上显示出输入的化学分子式中的原子数

"""

```
formular = input("请输入化学分子式：")  
print(f"{formular} 中的原子数量为 {atom_count(formular)}")
```

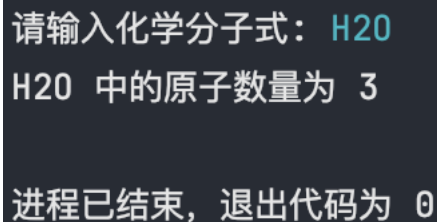
```
if __name__ == '__main__':  
    main()
```

运行结果：

请输入化学分子式：H2O

H2O 中的原子数量为 3

进程已结束，退出代码为 0



请输入化学分子式：H2O
H2O 中的原子数量为 3

进程已结束，退出代码为 0

四. 实验总结

通过本次实验，我了解了词法分析和语法分析的基本原理和实现方法，同时掌握了使用 PLY 库构建解析器的基本技能。

实验中，我学会了如何设计和实现词法分析器和语法分析器，以及如何定义和使用文法规则来解析化学分子式。

通过编写和运行测试用例，我了解了如何验证解析器的正确性，以及如何排查和解决解析过程中可能出现的问题。

本次实验加深了我对编译原理课程内容的理解，为我在编译器设计和相关领域的进一步学习和探索奠定了基础。