

# 化学反应自动生成系统 人工智能专家系统

系 别：计算机科学技术学院

专 业：计算机科学与技术专业

学 号：17307130178

姓 名：宁晨然

指导教师：危辉

2019 年 12 月 29 日

## 摘要

本论文实现了化学反应方程式自动生成的专家系统，力图使用专家系统解决中小学生对无机化学不熟悉化学反应过程的问题。专家系统从化学反应池、合成化学路径和特定属性化学反应三个角度出发，解决了相应问题。并且实现了用户友好交互的前端，提供给用户询问相关化学问题，以获取答案。

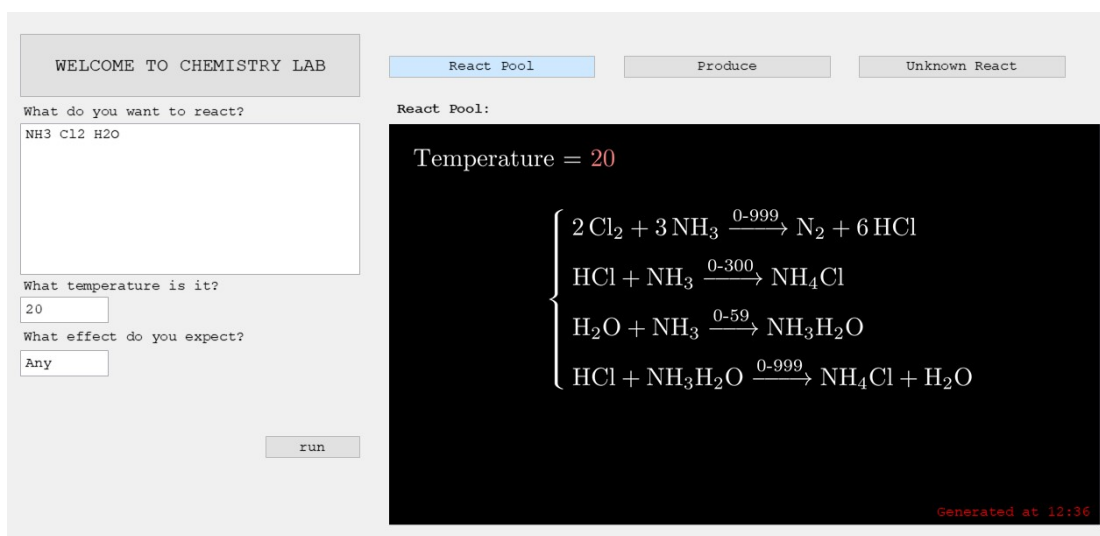


图1 氨气、氯气和水反应

本文的创新点主要有：

- 大量化学方程式的知识库<sup>[1]</sup>；
- 利用前向后向链接<sup>[2]</sup>推理触发推理链；
- 用 manim<sup>[3]</sup>生成用户友好的化学反应动画。

**关键词：**人工智能；专家系统；化学反应；无机化学；化学反应池

## 目 录

第 1 章 引入 .....	1
1.1 背景 .....	1
1.2 简介 .....	1
第 2 章 相关知识 .....	2
2.1 基于规则的专家系统 .....	2
2.2 知识库建立 .....	3
2.3 基于框架的专家系统 .....	4
2.4 前向链接与后向链接 .....	5
2.5 其他设计理念 .....	5
2.6 专家系统构架 .....	6
第 3 章 建立专家系统 .....	7
3.1 化学反应池 .....	7
3.2 化学合成 .....	9
3.3 特定属性的化学方程 .....	11
第 4 章 成果展示 .....	14
4.1 化学反应池案例 .....	14
4.2 化学合成、特殊方程案例 .....	16
第 5 章 评价 .....	17
参考文献 .....	19

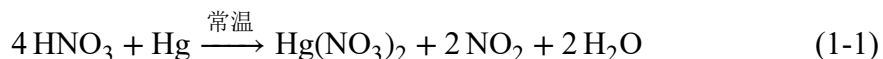
# 第 1 章 引入

## 1.1 背景

在初高中的化学课学习过程中，时常会遇到以下问题：给定几个反应物，写出反应物之间的所有化学反应方程式；给定某个生成物，写出合成生成物的可能的路径；已知化学反应方程式几个属性，写出可能满足条件的化学反应方程式。在无机化学的初学阶段，常常因为化学式的繁琐复杂、难以记忆，导致记忆不清化学反应方程式。然而网络上解答众多，答案繁多不一致反而给初学者带来极大困惑；更有同学羞愧于发问，不敢询问老师，造成知识盲点。此时他们需要诉诸一个化学反应方程式的专家系统，能够智能地解答他们以上疑惑。

## 1.2 简介

化学反应自动生成专家系统旨在解决初高中无机化学初学者的疑问，主要解决了三个问题：1. 向反应池中添加反应物发生的所有反应方程式 2. 合成某个化合物的所有可能途径 3. 寻找带有特殊属性的化学反应方程式。用户可通过前端操作，类似向老师发问一般，专家系统会显示结果化学方程式 1-1。



专家系统的特点在于在狭窄领域提供专家水平的答案解决问题，化学方程式生成契合度较高。化学反应自动生成专家系统能够解决无机化学初学者大部分问题，也能帮助他们更快速入门化学领域、记忆基本化学反应方程式。经过改进的该专家系统有实力能成为中学生必要的学习小程序。专家系统会显示出反应方程式与反应条件。

代码部分：1. 前端使用 pyqt 和 manim<sup>[3]</sup> 写成。2. 后端专家系统使用 python 写成。3. 使用 L<sup>A</sup>T<sub>E</sub>X 写成论文<sup>[4]</sup>。

## 第 2 章 相关知识

### 2.1 基于规则的专家系统

化学反应自动生成专家系统的核心内容在于基于规则的专家系统，因为化学反应方程式使用规则表示是自然的知识表达方式。基于规则的专家系统可以分为知识库、数据库、推理引擎、解释设备和用户界面。

首先，建立知识表达技术，规则<sup>[2]</sup>。分析化学方程式的数据结构，如公式 2-1。可知一个化学方程式就类似规则，由反应物  $\text{H}_2$  和  $\text{O}_2$  在条件  $T = 26\sim 300$  下生成  $\text{H}_2\text{O}$ ，这是 IF-THEN 结构的产生式规则。



将化学方程式抽象为产生式规则如公式 2-2



规则的前项 IF 的对象为多个反应物、反应条件、反应特性等化学反应方程式的属性。其中前项反应物的值为化学式类，反应条件有温度，反应特性有固体、液体、气体等。规则的后项 THEN 对象为生成物、反应效果。其中生成物为化学式类，反应效果有爆炸、气泡、燃烧等。如图 2.1 显示。

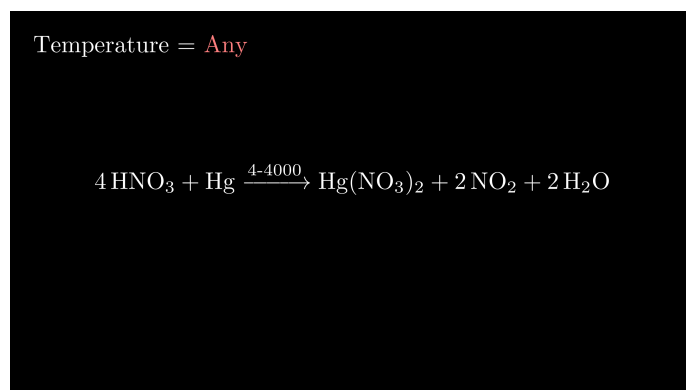


图 2.1 显示化学反应方程式

所以可以将化学反应方程式转换为规则。如表 2.1 所示。将化学反应方程式转换为规则之后需要涉及几个问题：前项化学式如何存储在知识库中；前项温

度、后项化学反应效果是附属哪个数据结构的属性；数据库中如何表示化学方程式的规则；生成物和反应物是否需要区分等。这就需要数据结构表征化学式、化学方程式和反应过程。

表 2.1 化学方程式对应的规则

前项	后项	规则
H <sub>2</sub>	H <sub>2</sub> O	IF H <sub>2</sub> + O <sub>2</sub>
O <sub>2</sub>		AND 26 ≤ T ≤ 300
T = 26~300		THEN Create H <sub>2</sub> O

## 2.2 知识库建立

其次，建立知识库和推理方法。本篇的数据库来源于软件烧杯 BEAKER<sup>[1]①</sup>。反汇编其 Andriod apk 文件获取其中化学式和反应方程的数据，表 2.2 摘选自其中的未处理数据。

cdb.csv 文件中存储了化学式对应的属性，有名称、状态、是否可获得、俗名、相对分子质量、密度等。rdb.csv 文件中存储了化学方程式对应的反应速度、反应物、生成物、配平系数、温度、状态、反应效果、反应颜色等。为了便于链式规则推理，创建类 *Chemical* 和类 *Equation*，使用全局变量 *data\_chemi* 和 *data\_equa* 存放所有数据。

表 2.2 未处理的 BEAKER 数据 Tab1: 化学式 Tab2: 化学方程式

Chemical	Type,State,Available,RName,CName,DName,Density1,Density2,Density3,Mm
H2O	Liquid,0,1,Water,H2O,H2O,0.917,1,0.001,18.02
Litmus	Liquid,4,1,Litmus,Litmus,Litmus,1,1,0.001,213.04
Ag	Solid,0,1,Silver,Ag,Ag,10.49,8.02,0.001,107.9
Al2O3	Solid,0,1,Aluminium oxide,Al2O3,Al2O3,3.95,1,0.001,102
Name	Speed,Reactants,Products,Modulus,Temps,States,Effects,Rek,Effect Color
Ag_Br2	0.1,Ag-Br2,AgBr,2-1-2,300-800,1-2,BURN3,1,0xFFFFFFFF
Ag_Cl2	0.3,Ag-Cl2,AgCl,2-1-2,90-900,1-3,BURN1,0.01,0xFFFFFFFF
Ag_F2	0.1,Ag-F2,AgF,2-1-2,229-800,1-3,BURN2,3,0xFFFFFFFF
Ag_H2SO4	0.3,Ag-H2SO4,Ag2SO4-SO2-H2,2-2-1-1-2,0-100,1-4,FALSE,0.01,0xFFFFFFFF

① IOS 和 Andriod 上开发的一款娱乐化学教育软件

该数据库较好，涵盖大部分无机化学的化学式和化学方程式。该数据库涵盖无机化学的大部分基础知识。而知识专家需要把这些原生知识生成存储在知识库中的规则，所以需要以下处理：1. 先建立纯净物化学式的知识库，作为先导知识，筛选出有用的属性信息。2. 再建立化学反应的推理知识库（规则），需要检验是否在纯净物化学式知识库中有对应信息，必要时新建纯净物。3. 验证知识库是否完备、可推理。

类 *Chemical* 和类 *Equation* 如表 2.3:

表 2.3 类 *Chemical* 和类 *Equation*

(a) 类 <i>Chemical</i>		(b) 类 <i>Equation</i>	
Chemical	H <sub>2</sub> O	name	H <sub>2</sub> _O <sub>2</sub>
Type	Liquid	reactants	[ <i>c</i> (H <sub>2</sub> ), <i>c</i> (O <sub>2</sub> )]
available	1	products	[ <i>c</i> (H <sub>2</sub> O)]
dname	H <sub>2</sub> O	modulus	[2,1,2]
color	OxFFFFFF	temperature	[26~300]
		effect	None
		color	OxFFFFFF
		equation	H <sub>2</sub> + O <sub>2</sub> $\xrightarrow{26\sim 300}$ H <sub>2</sub> O

知识库的评估见章节 5。

## 2.3 基于框架的专家系统

数据库建立后，用类的方式表达了知识库，自然而然想到使用基于框架的专家系统。这种系统强调面向对象，框架则是具有某个对象和典型知识的数据结构，此例中则是化学式。化学式代表了某个具有特定属性的纯净物（化合物和单质的总称），本专家系统中只有一个框架即化学式，化学式子类的实例则有各种反应物和生成物。

化学反应池中的框架则是：反应物、生成物、条件。框架使用 WHEN CHANGED 方法，即当反应物加入反应池中时，反应池激发链式规则发生化学反应并生成化学式。而当合成某个特定化合物时，框架使用 WHEN NEEDED 方法，即当所需化合物加入等待列表中，激发后向链接获得所需原料。

## 2.4 前向链接与后向链接

化学反应池和合成化合物分别使用了前向链接和后向链接。前向链接通过随机排列反应池中化学式，去匹配化学反应方程式数据库中的规则，如果符合 IF 事实（特定反应物和条件），则生成生成物，添加进入反应池（事实列表）。其中注意：

**定理 2.1：** 每个化学方程式只会触发一次

**定理 2.2：** 当化学方程式被触发，生成物加入反应池

**定理 2.3：** 没有化学方程式被触发时，反应池达到稳定

后向链接通过目标驱动，循环在规则数据库的化学方程式的 THEN 部分即生成物中寻找，并且判断 IF 部分的反应物是否可获得，不可获得则压栈，再寻找不可获得的反应物，直到所有反应物都可获得，输出期间所有化学方程式和反应物，即目标的合成路径。其中注意：

**定理 2.4：** 每个化学方程式只会触发一次

**定理 2.5：** 当化学方程式被触发，不可获得的反应物加入待获得列表

**定理 2.6：** 没有化学方程式被触发时，生成过程达到稳定

## 2.5 其他设计理念

还有需要解决的设计问题：

Q1: 为什么不使用模糊推理或者神经网络？

ANS: 因为化学反应本身是一个客观存在的事实，决定反应发生与否不是模糊的过程、也不需要进化算法的支持，当反应条件达成、反应物足够时就会触发反应规则，这符合规则推理的专家系统。

Q2: 知识库支持修改吗？

ANS: 可以。知识库是在 csv 文件上建立的，如果需要修改或者删减，直接在 csv 中修改对应的化学式或者化学反应即可。

Q3: 规则推理会冲突吗？



ANS: 不会。化学反应本身不是冲突的规则推理，而知识库中的知识先后顺序无关（无机化学比较简单），并且知识库中的反应只有一个或者两个反应物的情况，简化了推理达成条件。

Q4: 用户可以询问的范围？

ANS: 无机化学范围内的三百多个化学反应方程式。

Q5: 前端如何设计？ANS: 前端分为两个部分：1. 使用 QT 构成的专家系统交互系统，可以在系统中调整输入的参数。2. 点击 RUN 后会生成化学反应方程式的动画演示，见附件 1。

## 2.6 专家系统构架

化学方程自动生成专家系统架构如图 2.2

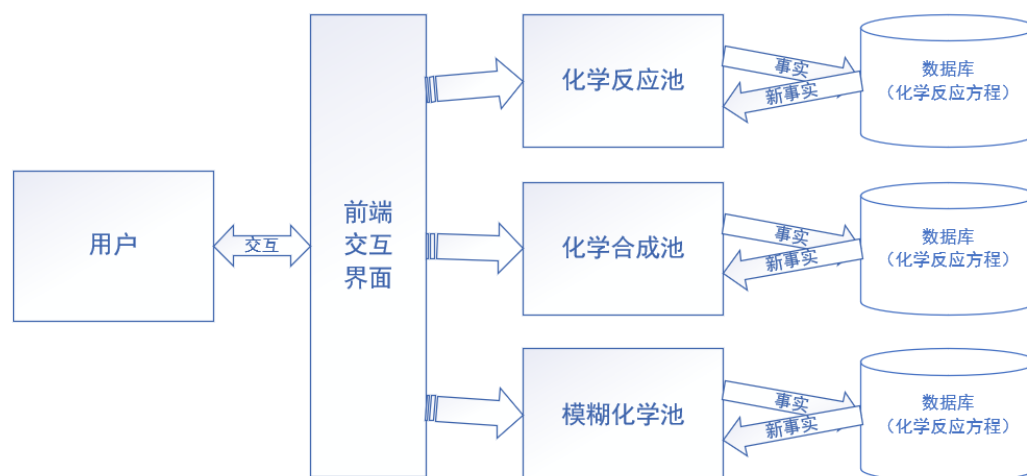


图 2.2 化学方程自动生成专家系统

## 第 3 章 建立专家系统

下面正文分三个板块介绍该专家系统，

### 3.1 化学反应池

生活中学习无机化学的过程中，经常会做化学反应池的实验。在某个烧杯中，加入多个液体、多个固体，观察反应发生的现象，推测出反应池中可能发生的所有反应方程式。所以，了解化学反应池中可能发生的反应就至关重要。

为了解决这个问题，首先建立数学模型。按照之前建立的规则推理引擎和知识数据库，需要建立一个化学反应池 `react_pool` 的框架如 3.1。化学反应池中需要包含属性值成分和方法：属性中，有反应物、生成物、反应条件等；方法中，有添加反应物、寻找反应方程式等。

表 3.1 框架 `react_pool`

属性	值
<code>reactants</code>	$\text{H}_2 + \text{O}_2$
<code>products</code>	$\text{H}_2\text{O}$
<code>temperature</code>	$26 \leq T \leq 300$
<code>effects</code>	<i>None</i>
<code>equation</code>	<code>[Equation(H2_O2)]</code>
<code>waitlist</code>	<code>[]</code>
<code>equation_str</code>	$2 \text{H}_2 + \text{O}_2 \longrightarrow 2 \text{H}_2\text{O}$
方法	作用
<code>get_init</code>	获取最初状态反应池的反应物、反应条件
<code>add_react</code>	添加多个反应物
<code>add_one_react</code>	添加一个反应物
<code>find_reaction</code>	寻找触发的规则
<code>general_fake_reaction</code>	列出可能触发的规则

最核心的算法是 `add_one_react`，添加反应物。添加多个反应物可以分批多次添加反应物，因为反应物添加顺序与最终反应池中发生的化学反应方程无

关。而添加单个反应物会触发 `general_fake_reaction` 列出可能触发的规则，利用 `find_reaction` 匹配规则。匹配一轮规则后，可能生成未在反应池中的反应物，则相当于又在反应池中添加了新的反应物，添加进入等待列表。每次从等待列表中取出一个反应物，利用 `add_one_react` 添加，触发规则结束后，循环从等待列表取出反应物，直到反应池稳定。用伪代码描述如表 3.2。

表 3.2 *React\_pool* 核心函数

(a) Function <i>add_one_react</i>	(b) Function <i>find_reaction</i>
Inputs: reactants	Inputs: reactants
Outputs: equations	Outputs: equations
function equations =	function equations =
while waitlist is not empty	generate_rule(reactants)
ready = waitlist.pop()	if products not in react_pool:
find_reaction(ready, reactants in	add it into waitlist
pool)	log equations

化学反应池的实质是向反应池一个一个添加反应物，根据前向链接生成所有反应物并记录生成过程。

**交互过程：**前端会询问用户想要设定的温度，想要的反应效果，和反应池中的反应物化学式，接下来会生成反应方程式的动画效果。

**生成结果：**前端显示出最后生成的化学反应方程式，如图 3.1。

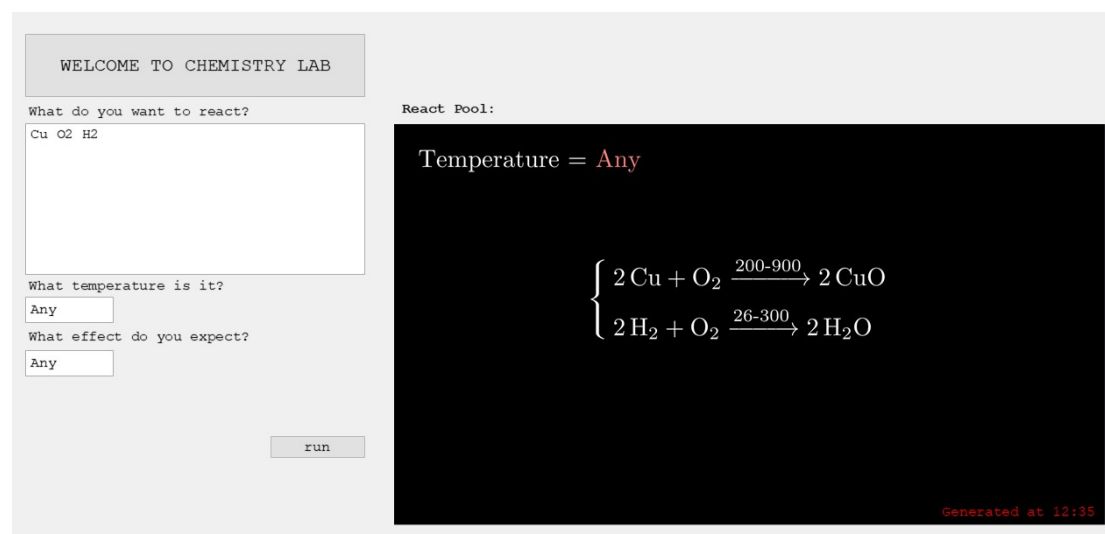


图 3.1

反应方程：生成反应方程式，如图 3.2。

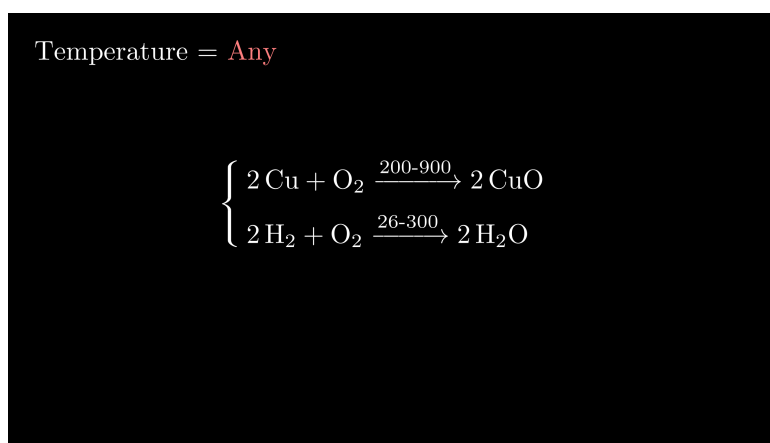


图 3.2

## 3.2 化学合成

生活中还需要解决合成某个特定化合物的情景，给定化合物、反应条件，写出反应的生成路径。由于很多化合物的属性值 `available` 为 `false`，即现实生活中无法直接通过购买得到，需要在实验室中通过实验生成。当然一个生成物可能有多个反应路径。建立数学模型，同理可以使用框架，推理引擎改为后向链接，建立一个化学合成池 `get_react` 的框架如 3.3。

表 3.3 框架 `get_react`

属性	值
<code>target</code>	<code>CsCl</code>
<code>temperature</code>	$T = 40$
<code>solutions</code>	<code>[Equation(Cs_H2O)]</code> <code>[Equation(CsOH_HCl)]</code>
<code>equation_str</code>	$2 \text{Cs} + 2 \text{H}_2\text{O} \longrightarrow 2 \text{CsOH} + \text{H}_2$ $\text{CsOH} + \text{HCl} \longrightarrow \text{CsCl} + \text{H}_2\text{O}$
方法	作用
<code>get_init</code>	获取最初状态的温度和目标产物
<code>add_target</code>	添加一个目标产物
<code>find_react</code>	寻找生成路径

最核心算法是 `find_react`，寻找合成路线。该算法是一个递归调用的后向链

接, 搜索生成目标产物的化学式: 如果反应物都 available, 则可以直接生成。(此处 available 是化学物的属性值), 如果反应物也不可获得, 则子程序调用 find\_react 合成。从反应物的所有合成路线中排列组合可以获得目标产物的合成路径。这样生成的合成路径很多, 选择其中一条显示。子目标物的规则触发完毕后, 即可合成父目标产物, 输出途中触发的所有规则即合成路径。伪代码如下:

表 3.4 核心代码 find\_react

---

```

Inputs: target
Outputs: solutions

```

---

```

Function solutions =
  If target in rule.then for rule in RULES:
    if reactants available
      produce solutions and return
    else
      child solutions = find_react(reactant)
      produce solutions with children
  return

```

---

合成目标产物的实质是对于某个特定的目标产物寻找可触发的化学方程式, 直到所需反应物都 available。

**交互过程:** 前端会询问用户想要设定的温度和目标产物, 选择最短合成路径, 接下来会生成反应方程式的动画效果。

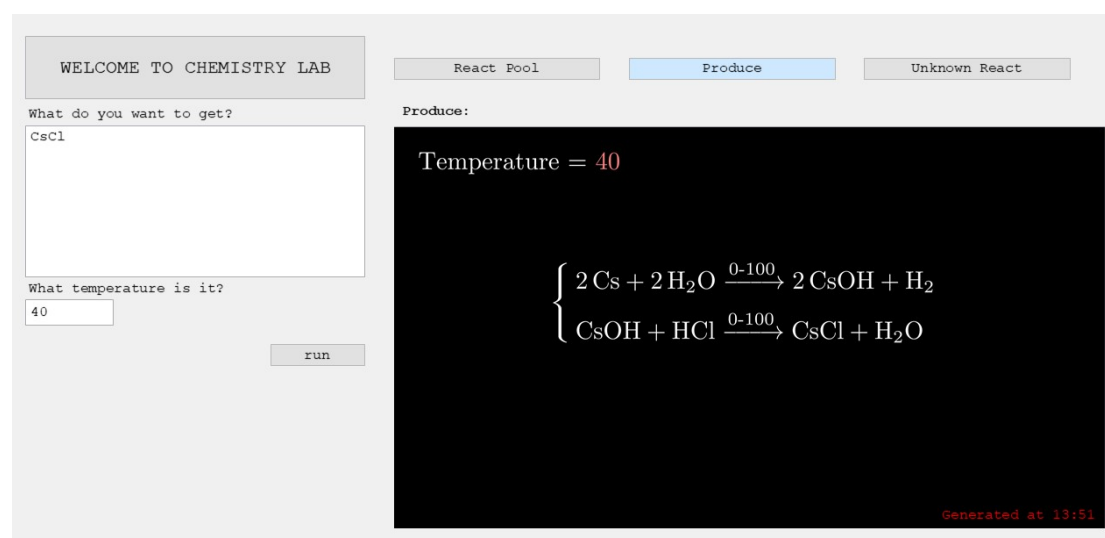


图 3.3

生成结果：前端显示出最后生成的合成路线，如图 3.3。

反应方程：生成合成路线反应方程式，如图 3.4。

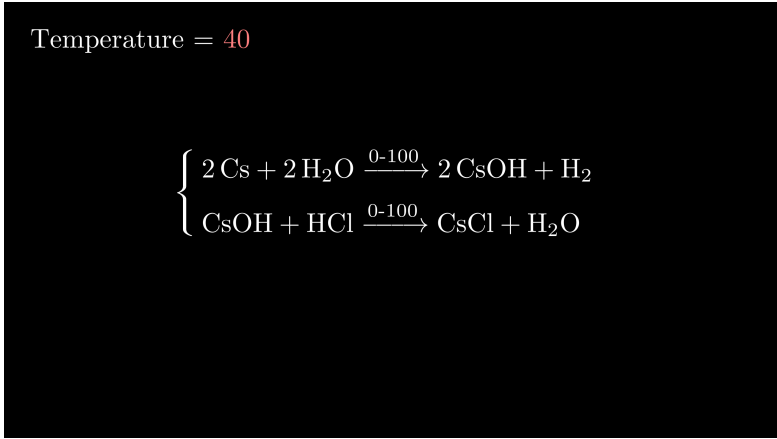


图 3.4

### 3.3 特定属性的化学方程

生活中第三个问题就是已知化学方程某些属性，比如“铜和液体生成气体的反应是什么？”，寻找对应方程式。由于这些属性既属于化学式的属性也是化学方程式的特征，所以建立数学模型同理使用框架，推理引擎为前向链接，建立一个特定属性的化学方程框架 state\_react 如表 3.5。

表 3.5 框架 state\_react

(a) 属性与值		(b) 方法与作用	
属性	值	方法	作用
reactants	<i>Cu Liquid</i>	get_init	获取最初状态的
products	<i>Gas</i>		温度和方程特征
temperature	$T = 100$	find	遍历知识库
effects	<i>any</i>	test_equ	判断是否触发
possible	[ <i>Equation(Cu_H2SO4)</i> ] [ <i>Equation(Cu_HNO3)</i> ]		
equation_str	$\text{Cu} + \text{H}_2\text{SO}_4 \longrightarrow 2 \text{H}_2\text{O} + \text{SO}_2 + \text{CuO}$ $\text{Cu} + 4 \text{HNO}_3 \longrightarrow \text{Cu}(\text{NO}_3)_2 + 2 \text{NO}_2 + 2 \text{H}_2\text{O}$ $3 \text{Cu} + 8 \text{HNO}_3 \longrightarrow 3 \text{Cu}(\text{NO}_3)_2 + 2 \text{NO} + 4 \text{H}_2\text{O}$		

最核心算法是 `test_equation`，触发前向链接规则的函数。该函数是一个判断是否符合 IF 触发条件的函数，需要满足：1. 反应物和生成物中必须包含用户给定的特定的化学式，如 Cu。2. 反应物和生成物的属性需要满足条件，如反应物有液体，生成物有气体。3. 满足温度条件。输出所有触发的反应方程式，伪代码如下：

表 3.6 核心代码 `test_equation`

---

Inputs: reactants products temperature  
Outputs: equations

---

```
Function equations =
for rule in RULES
IF temperature OK
AND reactants in rule.if
AND products in rule.then
AND attributes OK
THEN log equation
return
```

---

特定属性的化学方程式的实质是寻找具有特定特征的化学方程式，直到无规则可触发。

**交互过程：**前端会询问用户想要设定的温度和化学方程式的特征，接下来会生成反应方程式的动画效果。

The screenshot shows a web-based chemistry lab interface. On the left, there's a sidebar with the title 'WELCOME TO CHEMISTRY LAB'. It contains four input fields: 'What do you want to react?' (with 'Cu Liquid' entered), 'What do you want to produce?' (with 'Gas' entered), 'What temperature is it?' (with '100' entered), and 'What effect do you expect?' (with 'Any' entered). Below these is a 'run' button. On the right, there are three tabs: 'React Pool', 'Produce', and 'Unknown React' (which is selected). Below the tabs, the text 'Unknown React:' is followed by a large black box containing the text 'Temperature = 100' and three chemical equations: 
$$\begin{cases} \text{Cu} + \text{H}_2\text{SO}_4 \xrightarrow{55-1000} 2\text{H}_2\text{O} + \text{SO}_2 + \text{CuO} \\ \text{Cu} + 4\text{HNO}_3 \xrightarrow{4-4000} \text{Cu}(\text{NO}_3)_2 + 2\text{NO}_2 + 2\text{H}_2\text{O} \\ 3\text{Cu} + 8\text{HNO}_3 \xrightarrow{20-4000} 3\text{Cu}(\text{NO}_3)_2 + 2\text{NO} + 4\text{H}_2\text{O} \end{cases}$$
 At the bottom right of the black box, it says 'Generated at 14:15'.

图 3.5

生成结果：前端显示出最后生成的合成路线，如图 3.5。

反应方程：生成合成路线反应方程式，如图 3.6。

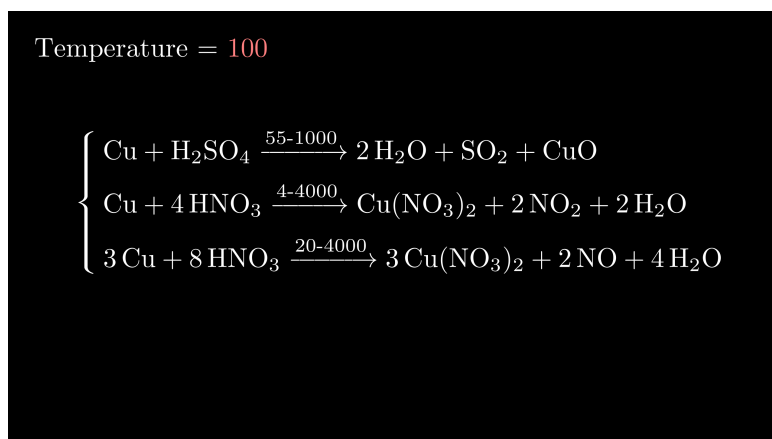


图 3.6



## 第 4 章 成果展示

### 4.1 化学反应池案例

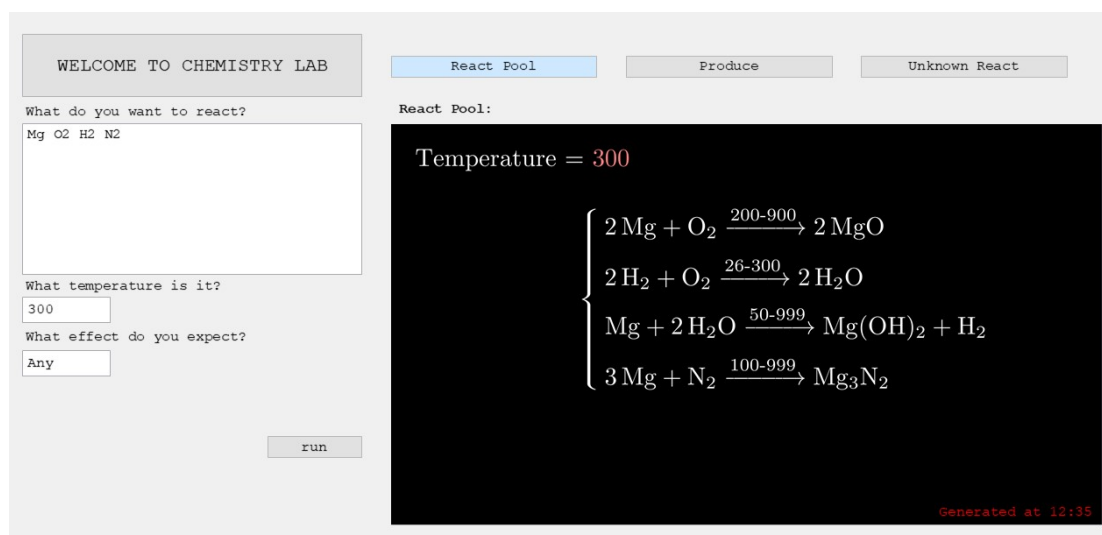


图 4.1 镁和氢氧氮气反应

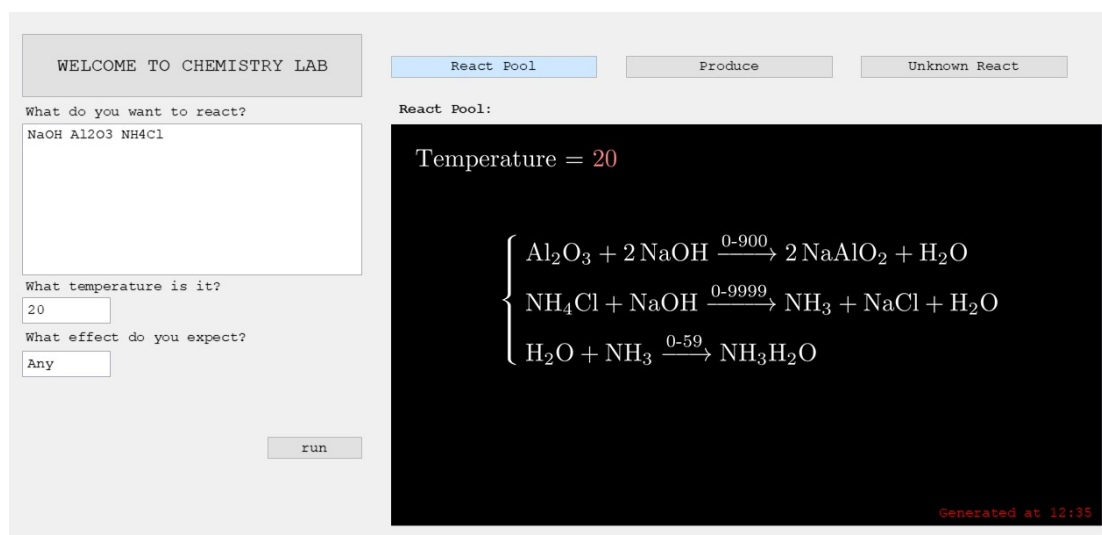


图 4.2 氢氧化钠、氯化铵和氧化铝反应

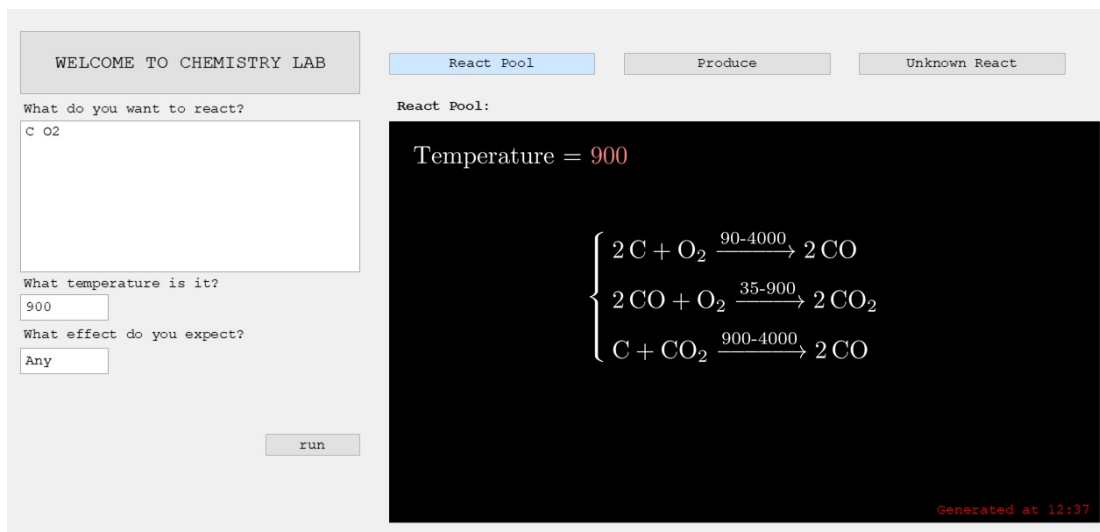


图 4.4 碳和氧气的高温反应

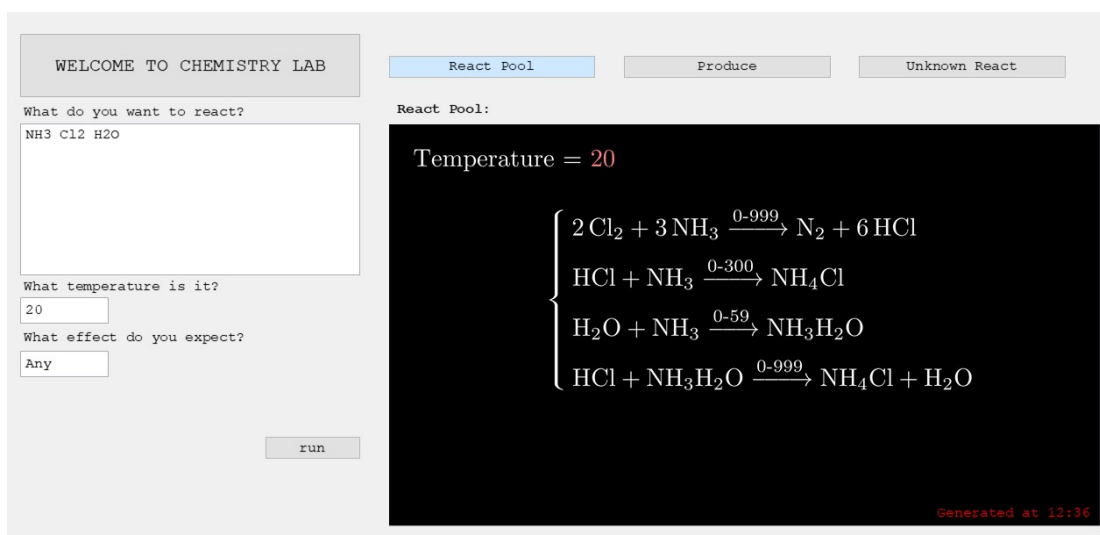


图 4.3 氨气、氯气和水反应

## 4.2 化学合成、特殊方程案例

The screenshot shows a web-based chemistry lab interface. On the left, a sidebar contains a 'WELCOME TO CHEMISTRY LAB' header, a text input field for 'What do you want to get?' with 'CsCl' entered, another text input field for 'What temperature is it?' with '40' entered, and a 'run' button. The main area has three tabs: 'React Pool', 'Produce' (which is selected and highlighted in blue), and 'Unknown React'. Below the 'Produce' tab, the text 'Produce:' is followed by a large black box containing the following information: 'Temperature = 40' in red, a set of chemical equations enclosed in large curly braces, and 'Generated at 13:51' in red at the bottom right. The equations are: 
$$\begin{cases} 2\text{Cs} + 2\text{H}_2\text{O} \xrightarrow{0-100} 2\text{CsOH} + \text{H}_2 \\ \text{CsOH} + \text{HCl} \xrightarrow{0-100} \text{CsCl} + \text{H}_2\text{O} \end{cases}$$

图 4.5 化学合成：合成 CsCl 过程

The screenshot shows the same chemistry lab interface. The sidebar inputs are: 'What do you want to react?' with 'Liquid Liquid', 'What do you want to produce?' with 'Gas', 'What temperature is it?' with '200', and 'What effect do you expect?' with 'Any'. The 'run' button is present. The main area has three tabs: 'React Pool', 'Produce', and 'Unknown React' (which is selected and highlighted in blue). Below the 'Unknown React' tab, the text 'Unknown React:' is followed by a large black box containing: 'Temperature = 200' in red, a set of chemical equations enclosed in large curly braces, and 'Generated at 14:17' in red at the bottom right. The equations are: 
$$\begin{cases} 3\text{H}_2\text{O}_2 + 2\text{NH}_3\text{H}_2\text{O} \xrightarrow{0-999} \text{N}_2 + 8\text{H}_2\text{O} \\ 3\text{H}_2\text{SO}_3 + 2\text{HNO}_3 \xrightarrow{0-1000} 3\text{H}_2\text{SO}_4 + 2\text{NO} + \text{H}_2\text{O} \\ 4\text{HNO}_3 + \text{Hg} \xrightarrow{4-4000} \text{Hg}(\text{NO}_3)_2 + 2\text{NO}_2 + 2\text{H}_2\text{O} \end{cases}$$

图 4.6 特殊方程：两液体混合生成气体的反应

## 第 5 章 评价

评价分为知识库、推理引擎和专家系统总体评价。

### 知识库的评价：

知识库主要是无机化学的化学反应方程式，则需要要求无机化学的知识体系健壮。知识库包括了 173 个化学物和 334 个化学方程式，涵盖了大部分无机化学反应，所以比较健壮，推理合理。其中常见的反应物和生成物（前十）如图 5.1和图 5.2所示。可知其中反应发生最多的是 $\text{HNO}_3$ ，生成最多的是 $\text{H}_2\text{O}$ ，符合化学常识。

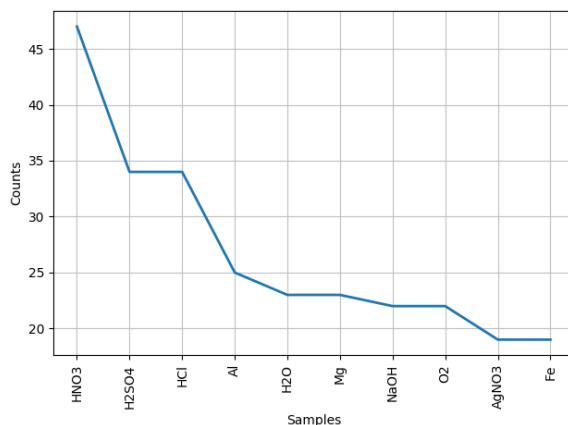


图 5.1 反应物频率图

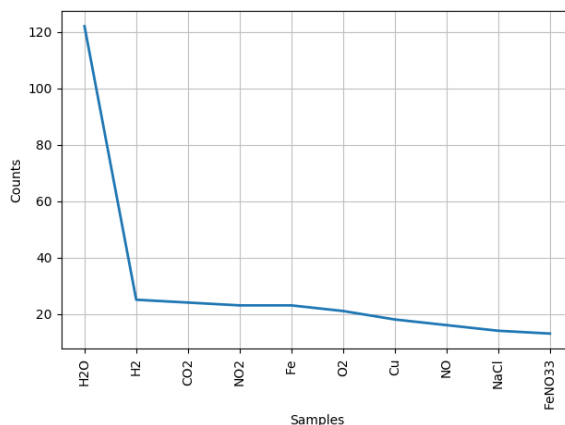


图 5.2 生成物频率图

知识库不足之处：1. 没有考虑化学反应的量的关系，因为化学反应因量不同可能发生的反应不同，因此也无法考虑反应速度。2. 无法考虑溶液反应中的酸碱性条件带来的反应环境不同，因为这也大量基于量。3. 无法建立金属活动顺序表和非活泼金属还原性等规则，这些规则的加入会使得专家系统异常复杂，需要专门的人工专家添加。

### 推理引擎的评价：

推理引擎与知识库独立，基本是镶嵌在三个框架中的。并且推理效率基本在  $O(N * e^k)$ ，所以很快能生成结果（下面推理时间不包括生成动画的时间）。计算出推理引擎 react\_pool 对于反应池中反应物的个数的推理时间，如图5.3。图是固定温度和反应效果均为 Any 不变，随机从知识库的化学式中选择固定多个反应物进入反应池，计算反应池生成化学反应方程式的总过程时间。由图可知前

向链接的推理引擎的时间效率随反应池中反应物个数增多而增多，且呈现指数级增长，但总体时间都在微秒级别，所以推理速度非常快！虽然没有人工标注的化学问题的样本集用于测试准确度，不过触发的化学式都是正确有效的。

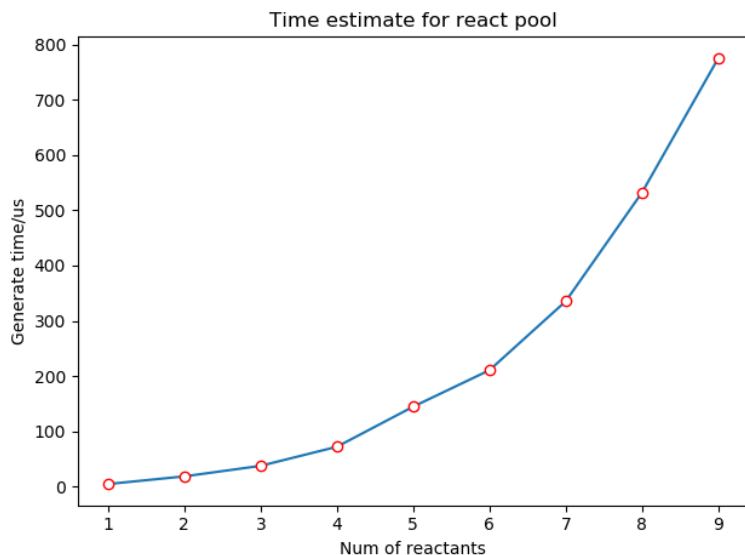


图 5.3

#### 专家系统总体评价：

该专家系统为中小学生初学化学提供了便利，解决了大量无机化学反应问题，有下面优点：1. 以可视化的动画形式教学化学反应过程，帮助用户记忆正确的化学方程式。2. 反应迅速，可以迅速给出答案。当然也有需要改进的地方：1. 没有建立完整的化学无机化学反应体系，没有考虑量和反应速度。2. 没有给出反应过程的图像展示。

总体而言，该化学反应专家系统较好的实现了教育专家的初衷，有一定功能实用性，也有许多待完善的地方。

#### 实验感想：

通过本次实验了解了专家系统的构造过程，并且动手实践提升了写代码的能力。虽然没有成功应用模糊逻辑和推理过程，但基于项目本身较好的实现了专家系统功能的完备性和准确性。

## 参考文献

- [1] THIX LLC. app: Beaker[EB/OL]. 2015. <https://apps.apple.com/cn/app/shao-bei/id961227503>.
- [2] Michael Negnevitsky. Artificial Intelligence: A Guide to Intelligence Systems, Third Edition [M]. New York: Pearson Education, 2012.
- [3] 3b1b. manim:animation engine for explanatory math videos[EB/OL]. 2018. <https://github.com/3b1b/manim>.
- [4] 薛瑞尼. ThuThesis: 清华大学学位论文模板[EB/OL]. 2017[2019-04-27]. <https://github.com/xueruini/thuthesis>.