

半机械化数学定理推导系统说明书

1 软件设计目的

该软件的主要目的是为数学专业师生及数学爱好者设计一个能摆脱繁琐纸笔演算的半自动化数学证明工具，根据数学定理的题设 (`given`) 和结论 (`imply`)，使用人机互动的算法，引导计算机求解数学证明题，最终使用 `latex`, `mathjax` 打印出数学命题证明的全过程。

2 软件主要功能

该公理化半机械化数学证明工具主要可以用于理论性数学证明，对数学学院的学生在学习和研究数学过程中有很高的参考作用，也可以用于数学教师整理数学定理知识，无需手动编辑繁琐的数学公式，也无需手动进行纸笔演算就可以完成数学知识的整理工作。对于研究和教学都有化繁为简的实用价值。主要功能在于三方面：

2.1 数据积累方面

提供了一个公理化的数学定理库，覆盖集合论，数论，代数，离散数学，组合学，几何学，三角学，微积分，概率论，深度学习等几个分支学科。

2.2 算法设计方面

提供了一个人机互动的机器证明算法，用户需引导机器调用已知定理，引导机器步步求解，机器自动打印证明过程，为用户节省了繁琐的纸笔演算时间。

2.3 图形用户界面

提供了一个可视化数学定理图形用户界面，包括关键词检索功能以及刻画定理之间依存关系的层级图，方便用户阅读相关数学定理。

3 软件实现原理

- 1, 首先对 [sympy](#) 符号计算算法进行改良, 使其能为数学定理推导服务。
- 2, 然后制定一套公理化的数学推导理论, 将所有数学命题使用严格的数学语法用公理, 定理, 运算法则推导出来, 在推导过程中不使用自然语言的辅助。
- 3, 实现一个半机械化的推导系统。每个数学证明题由三部分组成: 题设(given), 结论(implies), 证明过程(prove); 计算机的工作就是给出这个数学证明题的完整证明过程。如果计算机在证明过程(prove)中, 能根据题设(given)提供的信息, 使用已知数学定理库中的知识 (具体使用哪个定理由人脑决策给出), 对题设(given)中的条件表达式进行各种变换, 最终得出结论(implies)完全一致的条件表达式, 则表明计算机完成了给出了该数学命题的证明过程。
- 4, 人脑通过对已知定理库进行知识检索, 在推导过程发挥了数学定理选择决定权, 也就是告知计算机面对什么样数学命题使用什么样的已知定理进行下一步推理, 计算机根据人输入的 Python 命令利用符号计算算法和既定的逻辑推导法则进行推理, 得出一个更接近结论(implies)的表达式, 并最终得出结论(implies), 从而使数学命题获得证明。
- 5, 目前积累了约 850 个已知数学定理用于半机械化数学推导。主要涉及集合论(sets), 代数(algebra), 几何(geometry), 微积分(calculus), 数论和离散数学(discrete), 概率统计学(statistics), 深度学习中的数学模型(keras), 其中集合论是整个数学推导系统的理论核心。

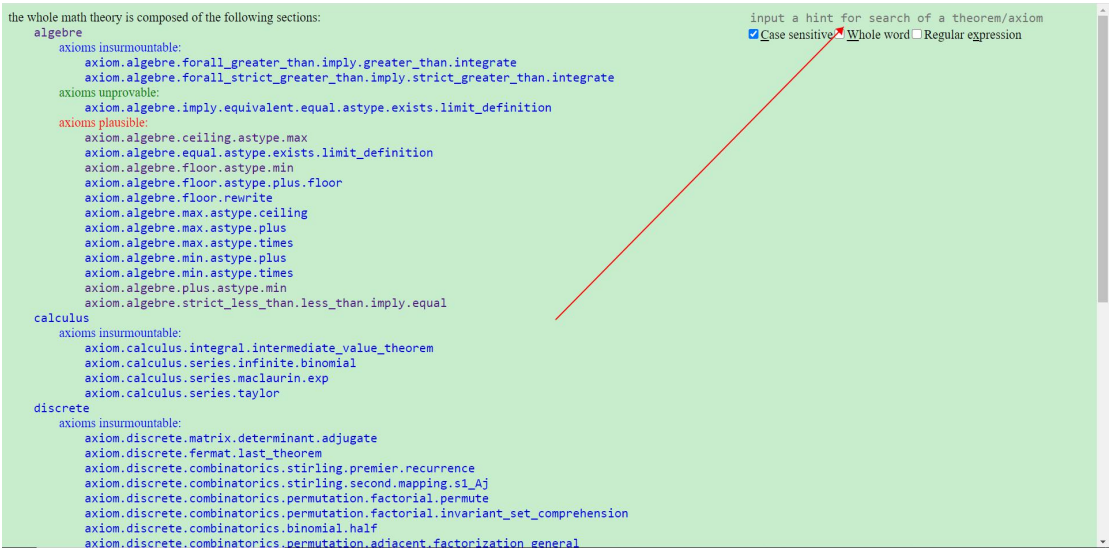
4 软件使用指南

4.1 在线使用指南

4.1.1 定理检索功能

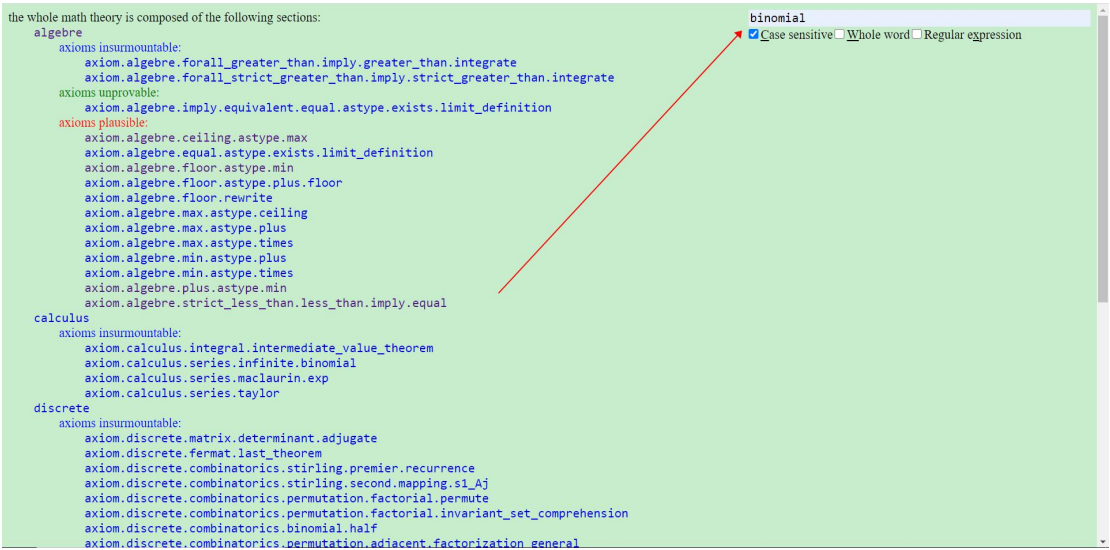
启动软件网站可视化定理库 <http://www.axiom.top/sympy/axiom/> (图 1.1),

或者定理库搜索界面 <http://www.axiom.top/sympy/axiom/search.php> 如下图所示:



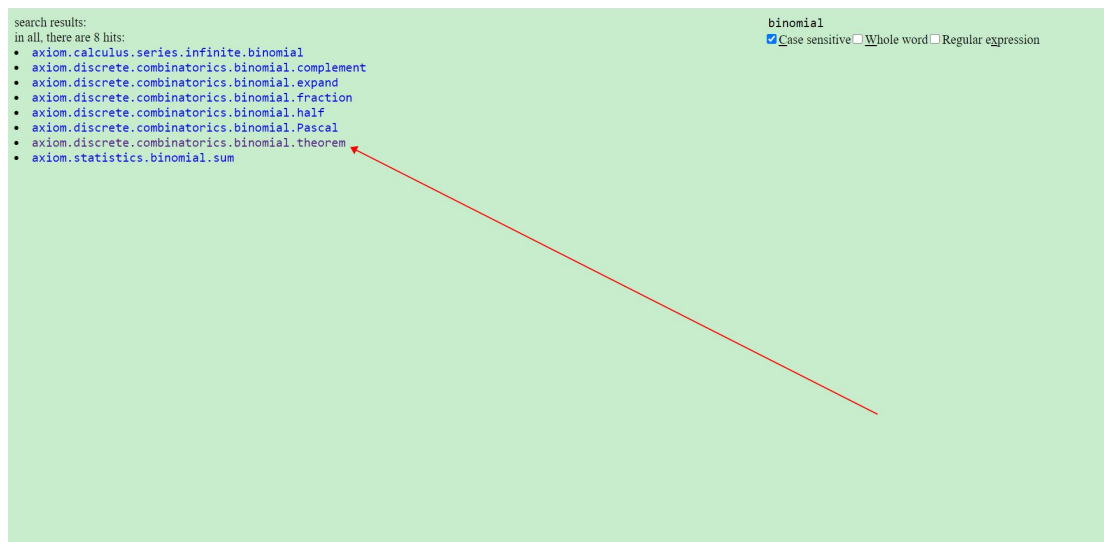
(图 1.1 数学定理库界面)

在提示符（input a hint for search of a theorem/axiom）中输入您想查找的数学定理名称，比如 binomial theorem(牛顿二项式定理) (图 1.2)



(图 1.2 关键词搜索主界面)

键入回车(Enter)之后，就可在前端打印出包含 binomial 单词的所有数学定理了，如下图 1.3 所示：



(图 1.3 关键词搜索结果显示界面)

按照箭头方向所指单击即可进入 [牛顿二项式定理](#) 的论证过程。

用户也可以直接访问 get api:

www.axiom.top/sympy/axiom/search.php?keyword=binomial

来获取搜索结果。

图形界面和 get api 同时也支持正则表达式以及全字(whole

word)匹配，比如：

www.axiom.top/sympy/axiom/search.php?keyword=discrete.*binomial®ularExpression=true

表示搜索在同时包含 discrete 和 binomial 单词的定理。

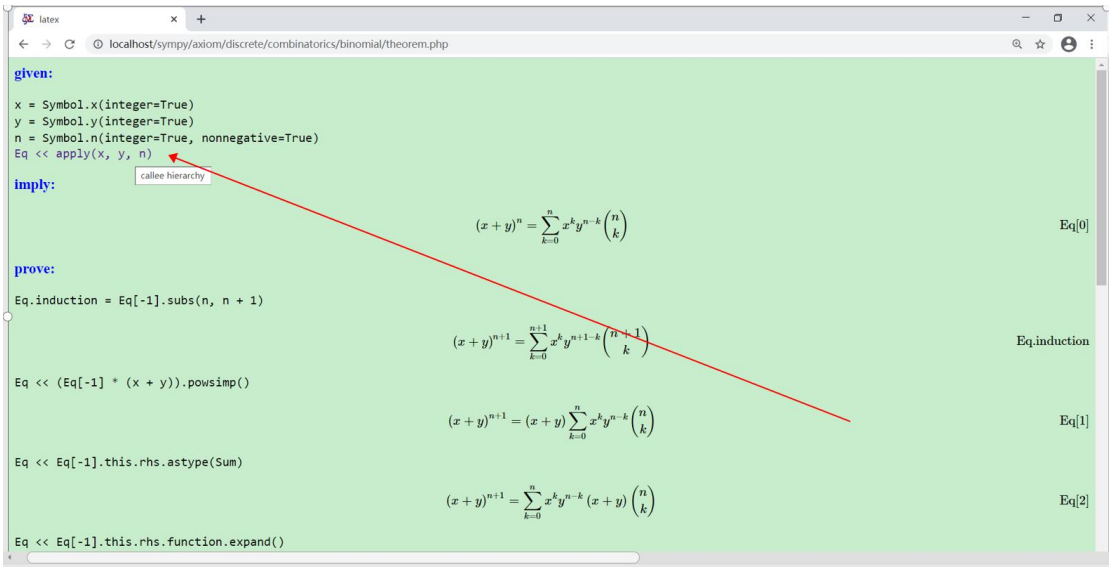
4.1.2 定理依存关系分析

4.1.2.1 上层定理引用关系

以下以 [牛顿二项式定理](#) 为例：

鼠标指向网页第一个超链接，会出现悬浮提示“callee hierarchy”。如图

1.4 所示：



(图 1.4 超链接处的悬浮提示语)

单击该超链接即可进入[上层定理引用层级图](#)。如图 1.5 所示：



(图 1.4 上层定理引用层级图)

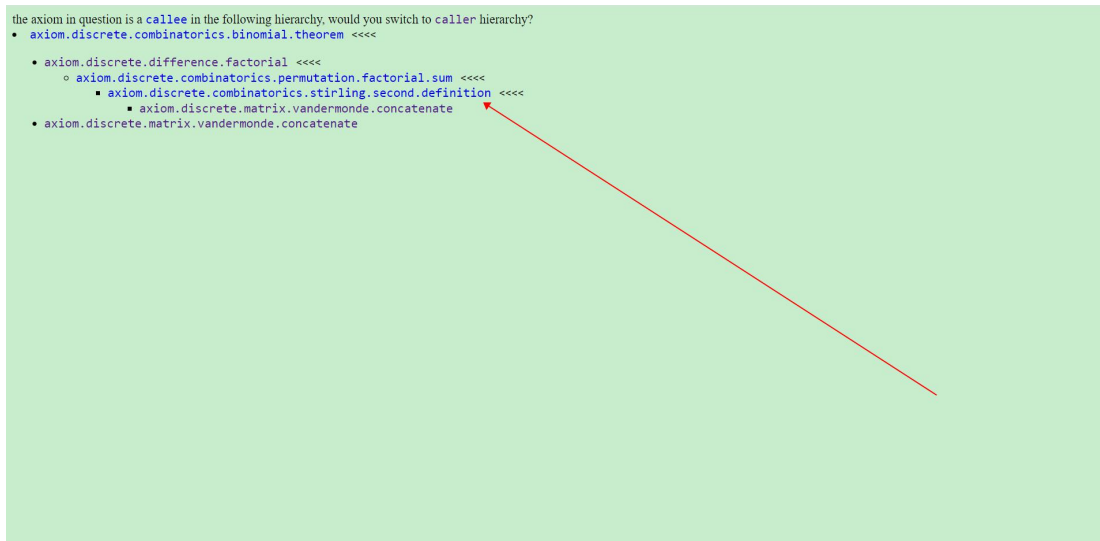
以上结果显示有以下几个定理在论证过程中引用了二项式定理：

[axiom.discrete.difference.factorial](#)

[axiom.discrete.matrix.vandermonde.concatenate](#)

在“上层定理引用层级图”中单击>>>(展开按钮),可以进一步查看上层定理的更上一层定理。单击<<<(隐藏按钮),可以隐藏展开后的定理。

在“上层定理引用层级图”中单击 [callee](#) 超链接,可以展开所有上层定理引用关系图。如图 1.5 所示：



(图 1.5 全部展开的上层定理引用层级图)

4.1.2.2 下层定理引用关系

在“上层定理引用层级图”中单击 [caller](#) 超链接，可以查看下层定理引用关系图。如图 1.6 所示：



(图 1.6 下层定理引用层级图)

以上结果显示在二项式定理的论证过程中引用了以下几个定理：

[axiom.discrete.combinatorics.binomial.Pascal](#) 组合数学中的 [Pascal 法则](#)，以法国数学家 Pascal 命名。

[axiom.algebre.sufficient.imply.condition.induction](#) 第一 [数学归纳法](#)，一

种递归证明方法。

在“下层定理引用层级图”中单击 [caller](#) 超链接，可以展开所有下层定理引用关系图。如图 1.7 所示：



(图 1.7 全部展开的下层定理引用层级图)

4.2 线下使用指南

线下使用该工具需要自己搭建本地 php 网站，具体 php 安装过程请参考安装部署文档 [php installation.docx](#)。线下使用还必须安装 python 开发环境。建议使用 python3.6 版的。安装完 python3.6 之后需要安装一个 python 依赖包：

```
pip install mpmath
```

以下介绍 python 工程的使用方法：

以线下 windows 版为例，指定一个网页文件夹，比如：E:\github，按照 [php installation.docx](#) 所提供的方法，修改 php 网站的 DOCUMENT_ROOT。然后进入该文件夹：

```
cd E:\github
```

使用 git 下载 python 工程源代码：

```
git clone --depth=1 https://github.com/cosmosZhou/sympy.git
```

进入 sympy 文件夹：

```
cd sympy
```

执行 run 命令，其中 debug=1 表示输出调试信息，debug=0 表示不输出调试信息：

```
python run.py debug=1
```

执行 run 完毕后会打印执行耗时，以及整个定理库的自动化证明的情况，比如：

```
in all 868 axioms
```

```
unproved:
```

(调试信息省略).....

```
websites:
```

(调试信息省略).....

```
seconds costed = 45.13455533981323
```

```
minutes costed = 0.7522425889968872
```

```
total unproved = 22
```

```
total failures = 0
```

最后打开 Chrome 或者 Edge/IE 浏览器，在浏览器地址栏输入：

<http://localhost/sympy/axiom/>

即可启动软件网站可视化定理库（如图 1.1 所示），这样就可以在本地环境下直接访问可视化定理库了。