MO GeoProver

MOGP Documentation

Debug 0.3.1

MOGP Development Team

September 30, 2022

MO GeoProver

MOGP 文档

编辑版 0.3.1

MOGP 开发团队

2022 年 9 月 30 日

前言

欧几里得的生平已无法考证，只知他约生于公元前330年，死于公元前275年。他将当时人们所取得的丰富知识，总结成几个原始概念和公理，并用逻辑推理的方法演绎成具有较为完整科学体系的几何学，完成了世界上第一部公理法巨著《几何原本》。也正因如此，欧几里得的名字与几何学成了同义词而流传千古。在几千年后的今天，现代公理化的欧几里得几何被广泛应用于中学教育中，以此锻炼学生的逻辑推理能力和想象力。与大多数人的认知不同，欧式几何其实是个难学的科目，因为欧式几何公理体系本质上是一种非机械化的数学：即使我们熟知了所有的公理，在解决问题时，依然要依靠人类的直觉和经验选择合适的定理求解问题，没有固定的解题步骤。

用固定的程序去解决一类问题，就是**机械化**。计算机出现后，人们尝试使用计算机实现欧式几何机械化，但长久以来该领域受制于**几何证明无定法**和**证明过程不可读**两大难题，进展缓慢。当今人工智能技术的快速发展，为几何问题机械化求解提供了新思路：使用**人工智能模型**模拟人的直觉来指导搜索和解题方向，以此解决几何证明无定法难题；设计一套精密且直观的**形式化语言**，作为计算机与人类沟通的桥梁，以此解决证明过程不可读的问题。在这个新思路中，推理过程涉及三个体系：

第一个体系是面向人类的**公理体系**。公理体系是指由欧几里得巨著《几何原本》发展来的现代几何公理系统。在该体系中，几何体的性质、定义和公理等采用自然语言和图像描述，此外代数表达式和一些基本常识也融入其中。人类理解起来这些描述非常轻松，任何一个经过训练的中学生阅读相关描述后都能理解题目所表达的意思；但因为自然语言所具有的模糊性和不确定性，计算机无法按照特定的算法来处理这些描述。于是第二个体系——**形式化体系**出现了。形式化体系是人类与计算机交流的中介，在三个体系中具有最重要的地位，因为其：①需要将公理体系中文字信息和图像信息转化为统一的格式，并且不造成信息损失和歧义。②语句描述要准确，且需具有固定的语法结构，便于计算机处理。③要具有良好的可读性，以供人类理解和检验推理步骤。最后一个体系是**计算机体系**。在此体系中，几何问题的描述进一步抽象、映射为特定的数据结构，如集合、列表等；几何定理则被抽象成定义在这些数据结构上的数据操作，如增删改查。

MOGeoProver（Mathematical Olympiad Geometry Prover, MOGP）几何问题求解系统可大致分为几部分：①MOGP 形式化系统：将几何问题的文字和图像信息转化为统一形式化语言描述，方便计算机解析处理。②MOGP 推理器：识别和解析形式化语言，机械化推理和验证几何定理、问题。③MOGP 解析器：包含文字解析器和图像解析器两大部分，分别将几何问题描述的文字信息和图像信息转化为 MOGP 形式化语言。④MOGP 定理（序列）预测器：为 MOGP 推理器提供机械化解题的方向；其输入为题目条件和解题目标的形式化描述，输出为下一步推理要应用的定理。

目录

一、MOGP 形式化系统

在前言中我们提到，作为人与计算机（或者说推理器）之间的沟通桥梁，形式化体系具有重要的作用。这要求形式化语言一方面必须方便人类理解，符合人类的认知习惯;另一方面必须具有一定结构,足够精确,以确保计算机正确解析。除此之外，为了完整的保存题目信息，形式化语言必须具有强大的表示能力，特别是用文本语言表示图像的能力。本章介绍 MOGP 形式化系统的设计思路。

1.1 几何本体论

本体论（Ontology）是哲学的第二个基本论域，本体论研究现象背后的本体性的成因，让现象之所以如此的那些基础性的本体，和这些本体之间的关系。换句话说，本体论主要关心的是事物存在的最根本原因。

回到我们的几何学上来，几何里面最根本的东西是什么呢？平面几何、立体几何、解析几何；点、线、角、面、体；各种定理、性质判定……令人眼花缭乱，什么是最本质的呢？华罗庚院士曾说过：“数缺形时少直观，形少数时难入微，数形结合百般好，隔裂分家万事休。”几何学，就是研究数与形的科学。

数与形是人类认识世界的两个角度。形是对于在某种意义上具有共性的一类对象的抽象描述。拿“三角形”来说，它是对于由三条边构成的图形的抽象描述。数是对某个对象的具体描述，刻画对象的某些性质，比如三角形某边的长度，三角形的面积等。自然界中并不存在抽象的三角形“三角形”，所存在的是三边有具体长度的、三角有具体角度的“三角形”。形是事物抽象的、质的描述，数是事物具体的、量的描述。数与形相互支撑，是人类认知世界能力的基础。

至此，我们发现了几何学中的两个本体。第一个本体比较直观，即图形，我们称之为**实体（Entity）**，如点、线、三角形等；第二个本体是数，说的更容易理解一点，即图形的**属性（Attribute）**，如线的长度、角的大小、图形的面积等。

上述分析是孤立的、片面的，而现实世界是相互联系、相互作用的。图形之间的相互作用，可以称之为**实体关系（Relation）**，如平行关系定义了两个直线之间的相互作用。图形属性之间的相互作用，即“事物的量”之间的比例关系，可以用各种**代数式（Expression）**来表示。形与数是认识世界的两个角度，是“不同世界”的东西，因此图形和属性之间定义关系是没有意义的。至此，我们又找到两个本体。

但是，我们遗漏了重要的一点。现实世界是动态的，是实时变化的，而我们上述的分析是在静态视角下进行的。现实世界根据物理规则，从一个状态转化到下一个状态。几何世界中的转化规则是什么呢？那便是**定理（Theorem）**。定理定义了一系列的规则，在满足一定的实体关系和代数关系的前提下，生成新的实体关系和代数关系。

这便是 MOGP 形式化系统的 5 个本体：实体、属性、实体关系、代数式、定理。

1.2 MOGP 形式化语言：语法设计和谓词设计

本小节先简要介绍谓词逻辑的一般概念，之后结合上一节中得到的 5 个本体，完成 MOGP 形式化语言的语法设计和谓词设计。本小节简单介绍设计思路，MOGP 形式化语言的详细规则请查阅第五章数据集标注指南。

1.2.1 谓词逻辑的基本概念

谓词逻辑是命题逻辑的发展，表达形式更加灵活、丰富。在命题逻辑中，有命题 A“小张喜欢几何”; 如果又有新命题“小松喜欢几何”, 只能用命题 B 表示。在谓词逻辑中，可以定义命题 $\mathrm{R}(x,y)$ 为“x 喜欢 y”，则上述命题 A、B 可以分别表示为 $\mathrm{R}(小张,几何)$ 、 $\mathrm{R}(小松,几何)$ 。其中 R 称作谓词，x、y 称作个体词。

个体词可以分为个体变项和个体常项，个体变项一般用 x、y、z 表示，个体常项一般用 a、b、c 表示。所有个体构成的集合，称为个体域或论域，是个体变项的变化范围。

谓词描述个体的属性，以及个体之间的关系，如“小张是学生”，可表示为 $isStudent(小张)$ 。根据所需个体词的数量，可以将谓词分为一元谓词、二元谓词……谓词是个体到真值的映射。

函数是个体到个体的映射,函数的引入增强了谓词逻辑的表达能力。比如“小张的朋友是学生”可表示为 $isStudent(Friend(小张))$ 。其中 $Friend$ 是函数，$isStudent$ 是谓词。

量词用来对个体的数量进行约束，常用的量词有全称量词∀和特称量词∃。

联结词是命题逻辑的基本概念之一,指由已有的命题构造出新命题所用的词语，常用的联结词有∧、∨、→等。

谓词、函数、量词、连结词可以构成**命题**。比如命题“小张的朋友全是学生”可以表示为 $\forall x\left(isFriend(\mathrm{x}) \rightarrow isStudent(x)\right)$。

由已有命题推导新命题的过程称为**推理过程**。如“小松是小张的朋友”、“小张的朋友全是学生”可表示为:

$$isFriend(小松) \wedge \forall x\left(isFriend(\mathrm{x}) \rightarrow isStudent(x)\right) \Rightarrow isStudent(小松)$$

1.2.2 实体

实体用谓词表示，如点、线、图形，可表示为 Point(A)、Line(AB)、Triangle(ABC)。其实体词由构成图形的点组合而成，组合规则将会在 1.4 节介绍。实体与实体之间存在隐含的关系，如若有 Triangle(ABC)，则一定会有 Line(AB)、Line(BC)、Line(AC)，这将会形成一个实体派生树，具体将会在第二章详细介绍。

1.2.3 属性

实体的属性用函数表示，定义了实体到符号的映射。如 Length(Line(AB))定义了线 AB 到符号 ll_bc 的映射，推理器识别到语句 Length(Line(AB))将会自动生成符号 ll_bc。其他的函数还有角度(Measure)、面积(Area)以及一些特殊的函数如周长、高等。

1.2.4 实体关系

实体关系也用谓词表示，如平行可表示为 Parallel(Line(AB),Line(CD))。实体关系区别于实体的表示，实体的表示是一元谓词，实体关系是多元谓词。

1.2.5 代数式

代数式部分可简单分为运算定义和 Equal 谓词。MOGP 形式化语言定义了包括加、减、乘、除、三角函数在内的 10 个基本运算，采用函数表示，如 Add(a,3) 定义了符号 a 和实数 3 到符号 a+3 的映射。Equal 语句声明相等关系，如 Equal(a,b)。

1.2.6 定理

定理是 MOGP 形式化系统中的动态部分，与谓词逻辑中的命题相对应。如勾股定理可以定义为：

$$
\begin{aligned}
& RightTriangle(ABC) \wedge \\
& Equal(Length(Line(AB)), a) \wedge \\
& Equal(Length(Line(BC)), b) \wedge \\
& Equal(Length(Line(AC)), c) \\
& \rightarrow \\
& Equal(Add(Pow(a,2), Pow(b,2)), Pow(c,2))
\end{aligned}
$$

为了表达简练，我们省略了全称量词。

1.3 完备性理论

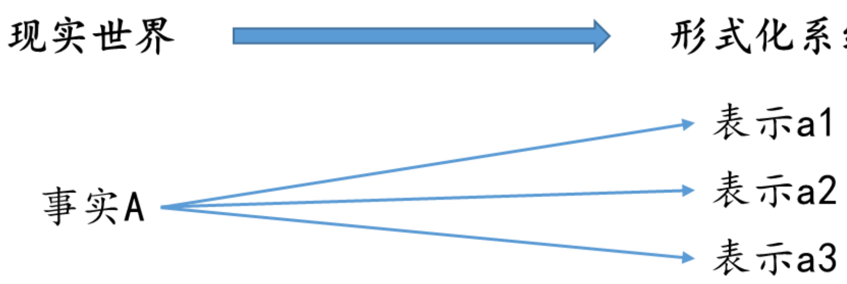
经过上述章节的讨论，我们完成了形式化语言语法设计和谓词设计。本章介绍完备性理论，探讨什么样的形式化系统是正确的、有效的，为下一章的个体词设计做好铺垫。

1.3.1 事实和表示

现实世界由各种**实体(Entity)**构成，实体本身具有各种**属性(Attribute)**，实体之间也会发生各种**关系（Relation）**。对于现实世界的实体、实体属性和关系的描述，称为**事实（Fact）**。现实世界时时刻刻都在发生变化，但不是随意变化。用本章语言来描述，即一些事实的集合，按照确定的转化规则，转化为另一些事实的集合，这两个事实的集合称之为**前提**和**结论**。

用于描述现实世界事实和转化规则的语言，我们称其为形式化系统。事实在形式化系统的描述，我们称其为**表示（Representation）**；事实和表示之间的映射规则，我们称之为**表示规则（Representation Rules）**；转化规则在形式化系统的描述，我们称其为**推理规则（Reasoning Rules）**。

形式化系统具有多态性和唯一性的特点，一个事实可能有多个表示，但一个表示只能唯一的确定一个事实。



1.3.1 多态性和唯一性

1.3.2 完备的定义

如果一个形式化系统的表示规则和推理规则能够正确模拟现实世界，则称该形式化系统是**完备（Complete）**的。

完备的充要条件：如果形式化系统的表示规则是完备的、推理规则是完备的，则称该形式化系统是完备的。

表示规则完备：①一个事实唯一的对应一组表示。②一个表示唯一的对应一个事实。

推理规则完备：一个推理规则由前提的表示和结论的表示构成，任意前提都能推出且只能推出结论的表示。

1.3.3 构造完备的形式化系统

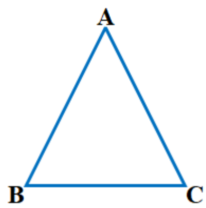
①设计完备的表示规则。②设计完备的推理规则。③在过程 2 中，若受制于表示规则的完备性而无法实现推理规则的完备性，则调整表示规则，直到表示规则和推理规则都满足完备性。

1.4 MOGP 形式化语言：个体词设计

1.4.1 单个图形的形式化表示

上一节介绍了完备性理论，一个正确的形式化系统应该建立正确的映射，不造成信息损失。几何问题的特点是既有文字描述，又有图像描述。在几何问题描述转化为 MOGP 形式化语言描述的过程中，文字信息是非常容易转化的，只需使其结构化即可，几乎不存在信息损失；对于图像信息的形式化，乍一想貌似需要考虑很多。接下来我们便分析如何将图形转化为形式化语言描述。

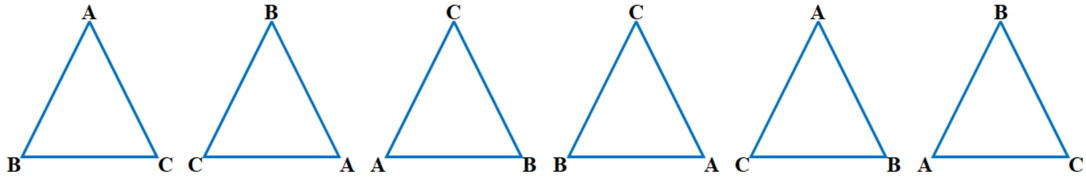
我们从 MOGP 形式化系统的 5 个本体（实体、属性、实体关系、代数式、定理。）入手，一步步确定图像形式化的关键。首先排除**定理**，静态的图像中不需要转化定理信息；其次再考虑**实体关系**，如果要在图上标明实体关系，就会使用特定的符号，如垂直、角相等，形式化也不会造成信息损失；然后考虑**代数式**，代数式描述的是属性值之间的比例关系，只能用文字描述，因为图像总会有误差，人在解题的时候也不可能用尺子去测量图形边的长度等；在后考虑**属性**，属性信息通常标记在对应实体的旁边，比如线的长度，形式化后也不会造成信息损失；最后就只剩下实体了。在去除了 4 个本体后，图像变得十分纯粹，我们从最简单的本体三角形入手分析。



1.4.1 三角形

如图 1.4.1 所示，我们看到一个三角形的图像。一个很直观的想法是把三角形的三个点作为图像的形式化描述，对三点排列组合后，我们得到 6 种表示形式

{ABC, BCA, CAB, CBA, ACB, BAC}。这样的形式化规则貌似是可行的，我们尝试将形式化描述还原成原始图像。根据完备性理论，如果可以唯一地还原成原图像，那我们的形式化规则就是正确的，无信息损失。

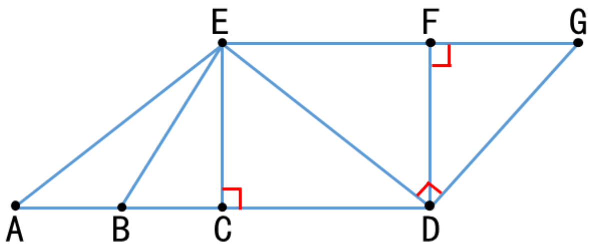


1.4.2 还原后得到的图形

糟糕，从形式化语言还原得到了 6 个图像。但仔细观察后发现前 3 个图形本质是相同的，只是进行了旋转操作，但旋转操作并不会改变图形的性质，同理后 3 个图形本质也是相同的。我们将其合并后，得到了一对镜像图形。为什么会这样？因为我们随手设计的形式化规则，在转化过程中，丢失了图形各点的相对位置信息。这也明确了，图像中有，而文本中没有的，就是各点之间的相对位置信息。涉及 3 个点及以上的图形，都需要保存点的相对位置信息。在 MOGP 形式化系统中，我们规定只有**逆时针排列的点的有序序列**，才是图形的合法表示，以此来保存点的相对位置信息，区分镜像图形。

1.4.2 组合图形的形式化表示

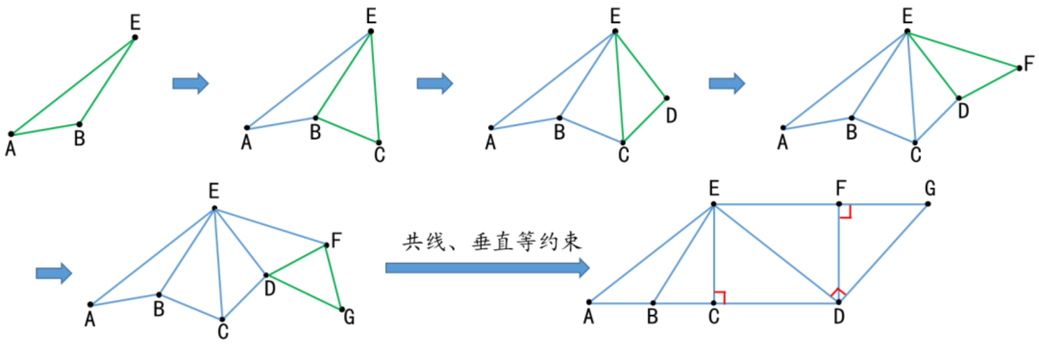
在上一小节中，我们探索了单个图形的形式化表示。然而在实际问题中，我们面对的是组合图形，如图 1.5.3 所示。3 个点是保存相对位置信息的最小单位，对于拥有 n 个点的图形来说，我们需要 $C_n^n+C_n^{n-1}+\cdots+C_n^3$ 条语句才能列举所有点的相对位置信息，当 n 增大时，语句的数量存在指数爆炸的风险。抛开这个问题不谈，有些点位于图形的内部（如重心），如何确定图形内部的点在相对位置表示序列中的位置，也是个令人头疼的问题。



1.4.3 组合图形示意

其实我们不需要 $C_n^n+C_n^{n-1}+\cdots+C_n^3$ 语句，就能保存所有点的相对位置信息，比如{EAB, EBC}中，蕴含了{EABC}。由 n 个图形组成的组合图形，我们只需要 n 条语句，便可以保存所有点的相对位置信息。图形可以经过组合得到新的图形，同样相对位置信息也可以经过组合得到新的相对位置信息，这也恰好吻合我们对于推理规则的完备性要求。

对于图 1.4.3，我们可以得到 5 条相对位置信息语句{EAB, EBC, ECD, EDF, FDG}。读取这 5 条相对位置信息语句，我们可以一步步的还原原始图形。最后添加共线、垂直等约束，就可以得到原始图形，我们称该方法为**拼图法**。



1.4.4 拼图法示意图

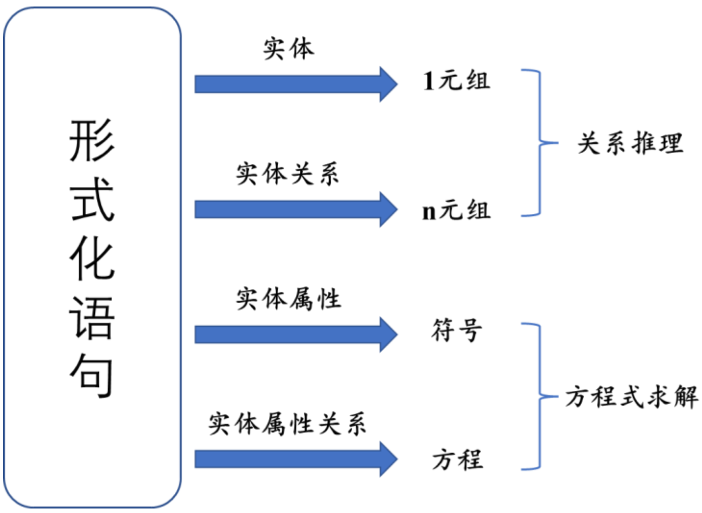
二、MOGP 推理器

MOGP 推理器是整个 MOGP 几何问题求解系统的中心。它一方面接受 MOGP 解析器的形式化语言输入，并解析成 MOGP 推理器内部的数据表示；另一方面接受 MOGP 定理（序列）预测器的输入，机械化的一步步执行定理。本章节简要介绍推理器的设计思想和特色功能。

2.1 推理器设计

本小节分为两个部分，第一部分介绍静态视角下，形式化语言的解析；第二部分介绍动态视角下，定理的实现。

2.1.1 形式化语句解析与数据表示



2.1.1 语句解析与数据表示

介绍四类本体的解析

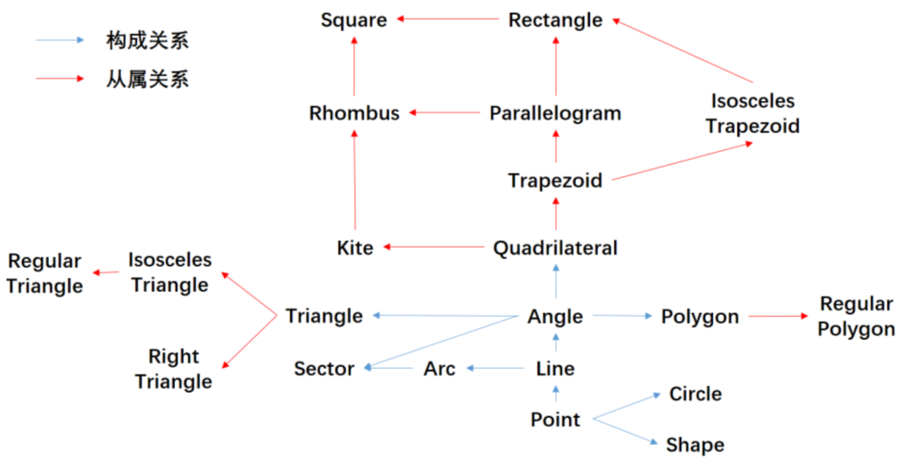
2.2.2 定理实现和方程式求解

两类解题目标 关系和数值
定理实现：满足前提的情况下推结论
方程式求解三步：①化简②寻找最小方程组③求解

2.2 特色功能介绍

2.2.1 条件自动扩展和派生

为什么自动扩展：模拟人解题时的常识，这些常识不会出现在解题过程，但是是解题必须的。



2.2.1 实体派生图

2.2.2 推理过程展示与回溯

可回溯的原因：index、premise 和 theorem

好处：①方便人检验，可解释②暴力搜索后去掉无用定理，防止训练预测器时引入噪声

2.3 例题演示

三、MOGP 解析器

3.1 文本解析器

rule-based or DeepNet-based?

3.2 图像解析器

图像分割、OCR、ViT or conv?

四、MOGP 定理（序列）预测器

4.1 MOGP 形式化语言 embedding

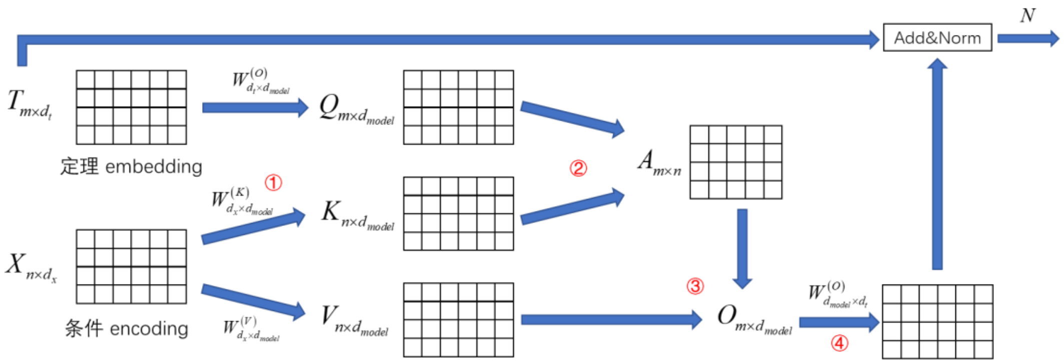
4.1.1 谓词 embedding

难点：embedding 依据

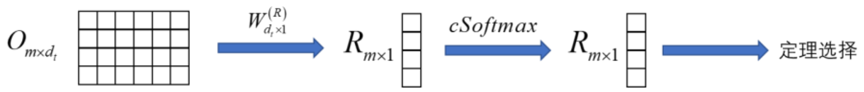
4.2.2 个体词 embedding

难点：embedding 依据、不定长序列的问题

4.2 基于注意力机制的定理（序列）预测器



① $K = XW$, $V = XW$, $O = TW$　② $A = rSoftmax\left(\frac{QK^T}{\sqrt{d_{model}}}\right)$　③ $O = AV$　④ $O = OW$



五、数据集标注指南

本章节对 MOGP 几何问题求解系统所有形式化语句的语法规则和标注方法做详细介绍，并配有适量的例题标注以方便读者理解。根据 1.2 节对 MOGP 形式化语言谓词的讨论，把形式化语句分为 6 类，分别是**构图语句、实体声明语句、实体属性声明语句、实体关系声明语句、代数表达式定义语句**和**解题目标声明语句**。

5.1 构图

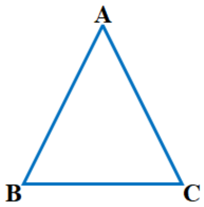
构图语句是最基本的形式化语句。推理器读取构图语句，自动扩展出所有的点、线、角和几何图形，实现 1.2 节中介绍的“拼图法”。

图形 Shape($)

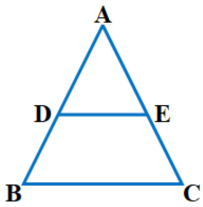
▸描述：内部无连线、由直线组成的封闭图形定义为 shape。shape 由若干 point 组成，shape 的标注要求按照逆时针按顺序标出所有的点。根据选择的起始 point 不同，一个 shape 可以有多种标注形式，选择一种即可。

▸要点：①内无连线的封闭图形 ②逆时针 ③多种形式选其一④标全

▸标注示例：



5.1.1 (a)



5.1.1 (b)

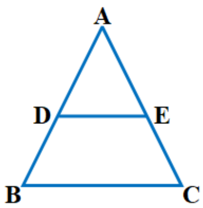
(a) Shape(ABC)
or {Shape(BCA)
Shape(CAB)}
(b) Shape(ADE), Shape(DBCE)

共线 Collinear($)

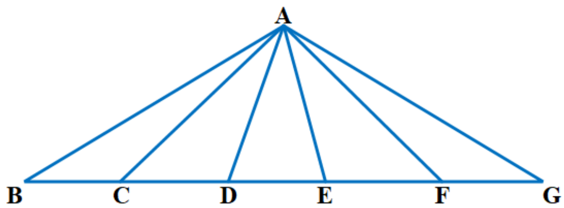
▸描述：共线的若干点定义为 collinear。collinear 由若干 point 组成，collinear 的标注要求按照一条线上点的先后顺序标出所有的点。根据选择的起始 point 不同，一个 collinear 可以有 2 种标注形式，选择一种即可。

▸要点：①共线点 ②先后顺序 ③2 种形式选其一④标全

▸标注示例：



5.1.2 (a)



5.1.2 (b)

(a) Collinear(ADB), Collinear(AEC)
or {Collinear(BDA)
Collinear(CEA)}
(b) Collinear(BCDEFG)
or {Collinear(GFEDCB)}

5.2 实体

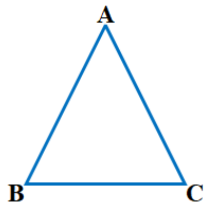
实体声明语句表示图形中的几何元素。

点 Point($)

▸描述：构成图形的点。可由推理器自动生成，无需标注。

▸要点：①无需标注

▸标注示例：



5.2.1 (a)

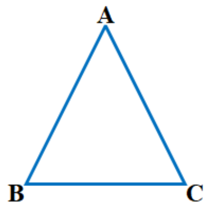
(a) Point(A), Point(B), Point(C)

线 Line($)

▸描述：构成图形的线。可由推理器自动生成，无需标注。一条线有两种标注形式，选择一种即可。

▸要点：①无需标注②2 种形式选其一

▸标注示例：



5.2.2 (a)

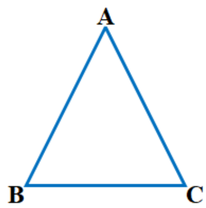
(a) Line(AB), Line(BC), Line(AC)

角 Angle($)

▸描述：构成图形的角。可由推理器自动生成，无需标注。一个角只有一种标注方式，逆时针标出构成角的三个点

▸要点：①无需标注②只有 1 种表示方式

▸标注示例：



5.2.3 (a)

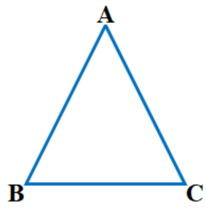
(a) Angle(ABC), Angle(BCA), Angle(CAB)

三角形 Triangle($)

▸描述：可由推理器自动生成，无需标注。逆时针标出构成三角形的三个点，因选择的起始点不同，一个三角形有 3 种表示方式。

▸要点：①无需标注②3 种形式选其一

▸标注示例：

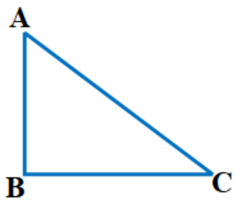


5.2.4 (a)

(a) Triangle(ABC)
or {Triangle(BCA)
Triangle(CAB)}

直角三角形 RightTriangle($)

▸描述：逆时针标出构成三角形的三个点。第二个点对应直角三角形中的直角，因此一个直角三角形只有一种表示形式。
▸要点：①逆时针②只有 1 种表示方式
▸标注示例：

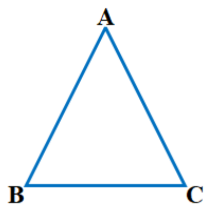


5.2.5 (a)

(a) RightTriangle(ABC)

等腰三角形 IsoscelesTriangle($)

▸描述：逆时针标出构成三角形的三个点。第二、三个点对应等腰三角形中的两个底角，因此一个等腰三角形只有一种表示形式。
▸要点：①逆时针②只有 1 种表示方式
▸标注示例：

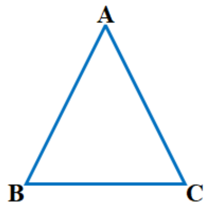


5.2.6 (a)

(a) IsoscelesTriangle(ABC)

等边三角形 EquilateralTriangle($)

▸描述：逆时针标出构成三角形的三个点。一个等边三角形有 3 种表示形式。
▸要点：①逆时针②3 种形式选其一
▸标注示例：



5.2.6 (a)

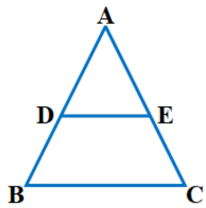
(a) EquilateralTriangle(ABC)
or {EquilateralTriangle(BCA)
EquilateralTriangle(CAB)}

多边形 Polygon($)

▸描述：边的数量大于 3 的图形称为多边形。可由推理器自动生成，无需标注。逆时针标出构成多边形的点。一个多边形有多种表示形式，选其一即可。

▸要点：①无需标注②逆时针③多种形式选其一

▸标注示例：



5.2.6 (a)

(a) Polygon (DBCE)
or {Polygon(BCED)
Polygon(CEDB)
Polygon(EDBC)}

5.3 实体属性

实体属性声明语句由一个谓词嵌套一个实体声明语句构成，表示实体的某个属性的度量。在形式化语句中实体属性声明语句不会单独出现，而是与 2.5 节中将要介绍的表达式定义语句嵌套使用。

在推理器中，每一条实体属性声明语句都会被转化为一个符号描述。

实体属性声明语句中的实体按照 2.2 中实体声明语句的标注标准标注。

线长度 Length(Line($))

▸描述：线的长度。一条线有 2 种标注方式，因此一条线的长度也有 2 种标注方式。

▸要点：①2 形式选其一

▸标注示例：线 DE 的长度

5.3.1 (a)

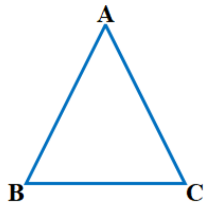
(a) Length(Line(DE))
or {Length(Line(ED))}

角大小 Measure(Angle($))

▸描述：角的大小。一个角只有 1 种标注方式，因此角的大小也有只有 1 种标注方式。

▸要点：①只有 1 种表示方式

▸标注示例：角 ABC 的大小



5.3.2 (a)

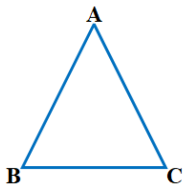
(a) Measure(Angle(ABC))

三角形面积 Area(Triangle($))

▸描述：三角形的面积。一个三角形有 3 种表示方式，因此三角形的面积也有 3 种表示方式。

▸要点：①3 种表示方式选其一

▸标注示例：三角形 ABC 的面积



5.3.3 (a)

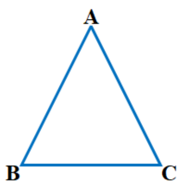
(a) Area(Triangle(ABC))
or {Area(Triangle(BCA))
Area(Triangle(CAB))}

三角形周长 Perimeter(Triangle($))

▸描述：三角形的周长。一个三角形有 3 种表示方式，因此三角形的周长也有 3 种表示方式。

▸要点：①3 种表示方式选其一

▸标注示例：三角形 ABC 的周长



5.3.4 (a)

(a) Perimeter(Triangle(ABC))
or {Perimeter(Triangle(BCA))
    Perimeter(Triangle(CAB))}

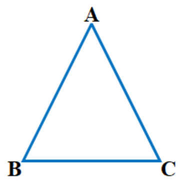
三角形高 Altitude(Triangle($))

▸描述：高的长度。特别提醒，一个三角形有三条高，每条高对应一种三角形的表示，因此一个高对应一个三角形表示。

▸要点：①只有 1 种表示方式

▸标注示例：

三角形 ABC 的高（底边 BC 的高）



5.3.5 (a)

(a) Altitude(Angle(ABC))

5.4 实体关系

实体关系声明语句说明图形中的几何元素之间的关系，这类关系通过定理可以推导出新的实体关系或实体属性的代数关系。

中点 Midpoint(Point($),Line($))

▸描述：点与线的关系。一条线有 2 种标注方式，因此中点也有 2 种标注方式。

▸要点：①2 种表示方式选其一

▸标注示例：

5.4.1 (a)

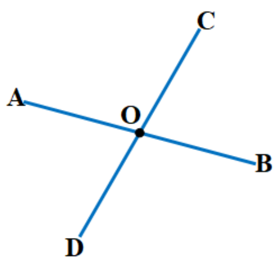
(a) Midpoint(Point(M),Line(AB))
or {Midpoint(Point(M),Line(BA))}

相交 Intersect(Point($),Line($),Line($))

▸描述：线与线的关系。其中 point 指两条线的交点；在标注两条线时，按照从左向右的顺序标注构成第一条线的点，按照从上到下的原则标注构成第二条线的点。一个相交关系有 4 种表示方法。

▸要点：①从左到右从上到下①4 种表示方式选其一

▸标注示例：



5.4.2 (a)

(a) Intersect(Point(O),Line(AB) ,Line(CD))

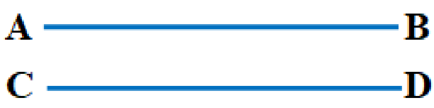
or {Intersect(Point(O),Line(CD) ,Line(BA)),
Intersect(Point(O),Line(BA) ,Line(DC)),
Intersect(Point(O),Line(DC) ,Line(AB))}

平行的 Parallel(Line($),Line($))

▸描述：线与线的关系。构成平行关系的两条线，其点的方向应是一致的，按照从左到右从上到下的原则标注。一个平行关系有 2 种表示方法。

▸要点：①从左到右从上到下①2 种表示方式选其一

▸标注示例：



5.4.3 (a)

(a) Parallel(Line(AB) ,Line(CD))

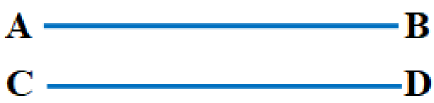
or {Parallel(Line(DC) ,Line(BA))}

无序平行 DisorderParallel(Line($),Line($))

▸描述：线与线的关系。构成平行关系的两条线，其点的方向应是一致的，按照从左到右的原则标注。无序平行不考虑两条直线的相对位置，一个相交关系有 4 种表示方法。

▸要点：①从左到右①4 种表示方式选其一

▸标注示例：



5.4.3 (a)

(a) Parallel(Line(AB) ,Line(CD))

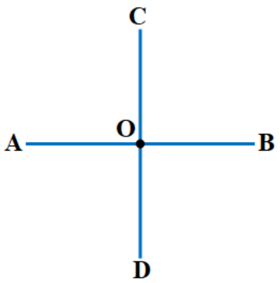
or {Parallel(Line(DC) ,Line(BA)),
Parallel(Line(CD) ,Line(AB)),
Parallel(Line(BA) ,Line(DC))}

垂直的 Perpendicular(Point($),Line($),Line($))

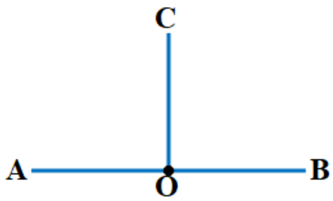
▸描述：线与线的关系。与相交的标注方法一致。需注意的是，垂直的图形有 3 种变形，如例题所示。

▸要点：①从左到右从上到下①4 种表示方式选其一

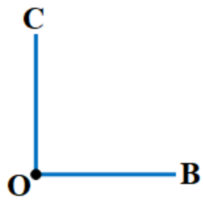
▸标注示例：



5.4.4 (a)



5.4.4 (b)



5.4.4 (c)

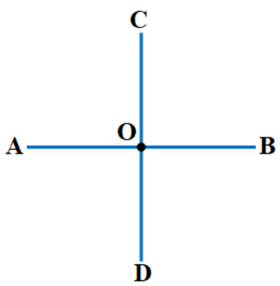
(a) Intersect(Point(O),Line(AB) ,Line(CD))
or {Intersect(Point(O),Line(CD) ,Line(BA)),
Intersect(Point(O),Line(BA) ,Line(DC)),
Intersect(Point(O),Line(DC) ,Line(AB))}
(b) Intersect(Point(O),Line(AB) ,Line(CO))
or {Intersect(Point(O),Line(CO) ,Line(BA)),
Intersect(Point(O),Line(BA) ,Line(OC)),
Intersect(Point(O),Line(OC) ,Line(AB))}
(c) Intersect(Point(O),Line(OB) ,Line(CO))
or {Intersect(Point(O),Line(CO) ,Line(BO)),
Intersect(Point(O),Line(BO) ,Line(OC)),
Intersect(Point(O),Line(OC) ,Line(OB))}

垂直平分线 PerpendicularBisector(Point($),Line($),Line($))

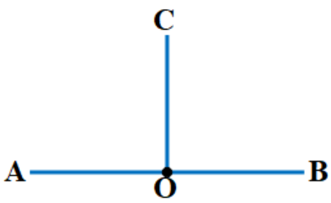
▸描述：线与线的关系。与垂直的标注方法一致。第一条线表示被平分的线，第二条线表示垂直平分线，因此此关系只有 2 种表示方式。垂直的图形有 2 种变形，如例题所示。

▸要点：①从左到右从上到下②2 种表示方式选其一

▸标注示例：



5.4.4 (a)



5.4.4 (b)

(a) Intersect(Point(O),Line(AB) ,Line(CD))
Intersect(Point(O),Line(BA) ,Line(DC))}
(b) Intersect(Point(O),Line(AB) ,Line(CO))

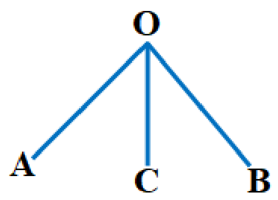
Intersect(Point(O),Line(BA) ,Line(OC))}

角平分线 Bisector(Line($),Angle($))

▸描述：线与角的关系。构成线的第一个点要与角的顶点相同。注意标注角时要按点的逆时针方向。

▸要点：①point 相同

▸标注示例：



5.4.5 (a)

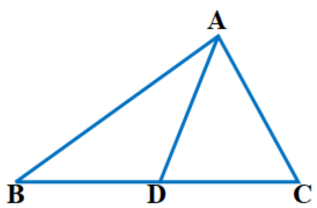
(a) Bisector(Line(OC),Angle(BOA))

中线 Median(Line($),Triangle($))

▸描述：中线是指三角形顶点与底边中点的连线。构成线的第一个点要与三角形的顶点相同。注意标注三角形时时要按点的逆时针方向。

▸要点：①point 相同

▸标注示例：



5.4.6 (a)

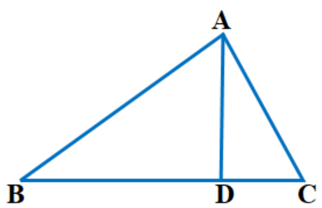
(a) Median(Line(AD),Triangle(ABC))

高 IsAltitude(Line($),Triangle($))

▸描述：三角形的高。构成线的第一个点要与三角形的顶点相同。注意标注三角形时时要按点的逆时针方向。

▸要点：①point 相同

▸标注示例：



5.4.7 (a)

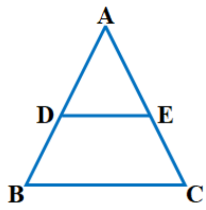
(a) IsAltitude(Line(AD),Triangle(ABC))

中位线 Neutrality(Line($),Triangle($))

▸描述：中位线是指平行于三角形底边，且位于三角形内部的线。线的方向与三角形底边方向相同。注意标注三角形时时要按点的逆时针方向。

▸要点：①方向相同

▸标注示例：



5.4.8 (a)

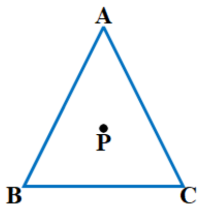
(a) Neutrality(Line(DE),Triangle(ABC))

外心 Circumcenter(Point($),Triangle($))

▸描述：三角形三边垂直平分线的交点。三角形有 3 种表示方式，因此外心也有 3 种表示方式。

▸要点：①3 种表示方式选其一

▸标注示例：



5.4.9 (a)

(a) Circumcenter(Point(P),Triangle(ABC))
or {Circumcenter(Point(P),Triangle(BCA)),
Circumcenter(Point(P),Triangle(CAB))}

内心 Incenter(Point($),Triangle($))

▸描述：三角形三角角平分线的交点。三角形有 3 种表示方式，因此内心也有 3 种表示方式。

▸要点：①3 种表示方式选其一

▸标注示例：

(a) Incenter(Point(P),Triangle(ABC))
or {Incenter(Point(P),Triangle(BCA)),
Incenter(Point(P),Triangle(CAB))}

重心 Centroid(Point($),Triangle($))

▸描述：三角形三边种线的交点。三角形有 3 种表示方式，因此重心也有 3 种表示方式。

▸要点：①3 种表示方式选其一

▸标注示例：

(a) Centroid(Point(P),Triangle(ABC))

or {Centroid(Point(P),Triangle(BCA)),
Centroid(Point(P),Triangle(CAB))}

垂心 Orthocenter(Point($),Triangle($))

▸描述：三角形高的交点。三角形有 3 种表示方式，因此垂心也有 3 种表示方式。

▸要点：①3 种表示方式选其一

▸标注示例：

(a) Orthocenter(Point(P),Triangle(ABC))

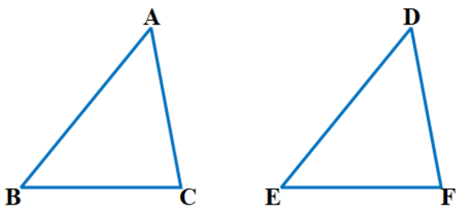
or {Orthocenter(Point(P),Triangle(BCA)),
Orthocenter(Point(P),Triangle(CAB))}

全等三角形 Congruent(Triangle($),Triangle($))

▸描述：按照点一一对应的方法，标注全等三角形。因为三角形有 3 种表示方式，两个三角形的先后顺序又可互换，所以全等有 6 种表示方式。

▸要点：①点一一对应②6 种表示方式选其一

▸标注示例：



5.4.10 (a)

(a) Congruent(Triangle(ABC),Triangle(DEF))

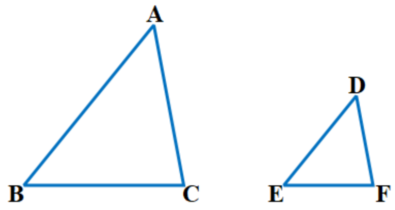
or {Congruent(Triangle(BCA),Triangle(EFD)),
Congruent(Triangle(CAB),Triangle(FDE)),
Congruent(Triangle(DEF),Triangle(ABC)),
Congruent(Triangle(EFD),Triangle(BCA)),
Congruent(Triangle(FDE),Triangle(CAB))}

相似三角形 Similar(Triangle($),Triangle($))

▸描述：按照点一一对应的方法，标注相似三角形。因为三角形有 3 种表示方式，两个三角形的先后顺序又可互换，所以相似有 6 种表示方式。

▸要点：①点一一对应②6 种表示方式选其一

▸标注示例：



5.4.11(a)

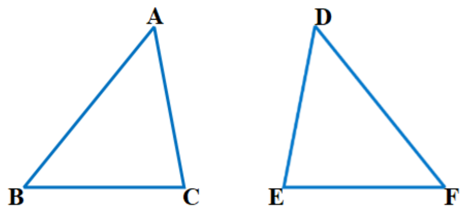
(a) Similar(Triangle(ABC),Triangle(DEF))
or {Similar(Triangle(BCA),Triangle(EFD)),
Similar(Triangle(CAB),Triangle(FDE)),
Similar(Triangle(DEF),Triangle(ABC)),
Similar(Triangle(EFD),Triangle(BCA)),
Similar(Triangle(FDE),Triangle(CAB))}

镜像全等三角形 MirrorCongruent(Triangle($),Triangle($))

▸描述：按照点一一对应的方法，标注镜像全等三角形。如例题所示，△ABC 按照点一一对应得到的全等三角形应是△DFE；但△DFE 是三角形的顺时针表示，不是合法的表示，因此交换后两点得到，△ABC 镜像全等△DEF。因为三角形有 3 种表示方式，两个三角形的先后顺序又可互换，所以镜像全等有 6 种表示方式。

▸要点：①点一一对应再交换②6 种表示方式选其一

▸标注示例：



5.4.12 (a)

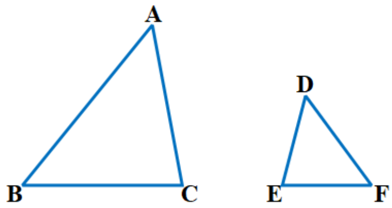
(a) MirrorCongruent(Triangle(ABC),Triangle(DEF))
or {MirrorCongruent(Triangle(BCA),Triangle(FDE)),
MirrorCongruent(Triangle(CAB),Triangle(EFD)),
MirrorCongruent(Triangle(DEF),Triangle(ABC)),
MirrorCongruent(Triangle(FDE),Triangle(BCA)),
MirrorCongruent(Triangle(EFD),Triangle(CAB))}

镜像相似三角形 MirrorSimilar(Triangle($),Triangle($))

▸描述：按照点一一对应的方法，标注镜像相似三角形。如例题所示，△ABC 按照点一一对应得到的相似三角形应是△DFE；但△DFE 是三角形的顺时针表示，不是合法的表示，因此交换后两点得到，△ABC 镜像相似△DEF。因为三角形有 3 种表示方式，两个三角形的先后顺序又可互换，所以镜像相似有 6 种表示方式。

▸要点：①点一一对应再交换②6 种表示方式选其一

▸标注示例：



5.4.13 (a)

(a) MirrorSimilar(Triangle(ABC),Triangle(DEF))
or {MirrorSimilar(Triangle(BCA),Triangle(FDE)),
MirrorSimilar(Triangle(CAB),Triangle(EFD)),
MirrorSimilar(Triangle(DEF),Triangle(ABC)),
MirrorSimilar(Triangle(FDE),Triangle(BCA)),
MirrorSimilar(Triangle(EFD),Triangle(CAB))}

5.5 表达式

代数表达式由变量、数字以及运算符三部分构成。例如在代数表达式 x+3 中，x 是变量，+是运算符，3 是数字。

在 2.3 节中提到过，实体的属性在推理器中会以符号（即变量）表示，因此实体属性声明语句可以与此节的代数表达式一起嵌套使用。本小节形式化语句中的参数 expr，可以由①代数表达式定义语句②代数表达式③实体属性声明语句

起始的 8 个谓词对应 8 种运算，分别是加、减、乘、除、幂、正弦、余弦、正切，对应运算符+、-、*、/、^、@、#、$。

加 Add(expr1,expr2)

▸对应表达式：expr1+expr2

▸示例： Add(x,3)
Add(Length(Line(AB)),3)
Add(Measure(Angle(ABC)),3+x)

减 Sub(expr1,expr2)

▸对应表达式：expr1-expr2

▸示例： Sub(x,3)
Sub(Length(Line(AB)),3)
Sub(Measure(Angle(ABC)),3+x)

乘 Mul(expr1,expr2)

▸对应表达式：expr1*expr2

▸示例： Mul(x,3)
Mul(Length(Line(AB)),3)
Mul(Measure(Angle(ABC)),Sub(Add(3+x,1),Length(Line(AB))))

除 Div(expr1,expr2)

▸对应表达式：expr1/expr2
▸示例： Div(x,3)
Div(Length(Line(AB)),3)
Div(Measure(Angle(ABC)),Sub(Add(3+x,1),Length(Line(AB))))

幂 Pow(expr1,expr2)

▸对应表达式：expr1^expr2
▸示例： Pow(x,3)
Pow(Length(Line(AB)),3)
Pow(Measure(Angle(ABC)),Sub(Add(3+x,1),Length(Line(AB))))

正弦 Sin(expr)

▸对应表达式：@expr
▸示例： Sin(x)
Sin(60)
Sin(Measure(Angle(ABC)))

余弦 Cos(expr)

▸对应表达式：#expr
▸示例： Cos(Add(x,y))
Cos(60)
Cos(Measure(Angle(ABC)))

正切 Tan(expr)

▸对应表达式：$expr
▸示例： Tan(x+3)
Tan(60)
Tan(Measure(Angle(ABC)))

实数 R

表达式 expr

8 种运算符、变量和数字的组合式子，需注意运算符之间的优先级，必要时可使用{}来规定各表达式之间的优先级，如{x+3}*2。

和 Sum(expr1,expr2, ...)

▸对应表达式：expr1+expr2+…
▸示例： Sum(x+3,y+2,z+1)

平均 Avg(expr1,expr2, ...)

▸对应表达式：{expr1+expr2+…+$expr_n$}/n
▸示例： Avg(x+3, y+2, z+1)

相等 Equal(expr1,expr2)

▸对应方程式：expr1-expr2=0
▸示例： Equal(x+3, y+2)
Equal(Measure(Angle(ABC)), 60)
Equal(Length(Line(AB)), z+4)

5.6 解题目标

Find 谓词用于定义解题目标，其参数可以是上述介绍的任意类型形式化语句，如实体声明语句，实体属性声明语句等。

Find($)

▸构图语句 自然语言：求证点 A、B、C 共线
形式化语言：Find(Collinear(ABC))

▸实体声明语句 自然语言：求证三角形 ABC 为等边三角形
形式化语言：Find(IsoscelesTriangle(ABC))

▸实体属性声明语句 自然语言：求边 AB 的长度
形式化语言：Find(Length(Line(AB)))

▸实体关系声明语句 自然语言：求证 AB 平行于 CD
形式化语言：Find(Parallel(Line(AB), Line(CD)))

▸代数表达式定义语句 自然语言：求 x 的值
形式化语言：Find(x)

2.7 定理

定理分为三类。第一类是常识(nous)，表示在题目中显而易见的条件，但是是解题所必须的；第二类是辅助作图(auxiliary)，暂时还没有定义相关类型的操作；第三类是定理。

nous_1_area_addition

nous_2_line_addition

nous_3_angle_addition

nous_4_

nous_5_

nous_6_

nous_7_

nous_8_

nous_9_

nous_10_

auxiliary_11_

auxiliary_12_

auxiliary_13_

auxiliary_14_

auxiliary_15_

auxiliary_16_

auxiliary_17_

auxiliary_18_

auxiliary_19_

auxiliary_20_

theorem_21_triangle_property_angle_sum

theorem_22_triangle_property_equal_line_angle

theorem_23_pythagorean

theorem_24_right_triangle_property_rt

theorem_25_right_triangle_property_special_rt

theorem_26_pythagorean_inverse

theorem_27_right_triangle_judgment

theorem_28_isosceles_triangle_property_angle_equal

theorem_29_isosceles_triangle_property_side_equal

theorem_30_isosceles_triangle_property_line_coincidence

theorem_31_isosceles_triangle_judgment_angle_equal

theorem_32_isosceles_triangle_judgment_side_equal

theorem_33_equilateral_triangle_property_angle_equal

theorem_34_equilateral_triangle_property_side_equal

theorem_35_equilateral_triangle_judgment_angle_equal

theorem_36_equilateral_triangle_judgment_side_equal

theorem_37_equilateral_triangle_judgment_isos_and_angle60

theorem_38_intersect_property

theorem_39_parallel_property

theorem_40_parallel_judgment

theorem_41_perpendicular_judgment

theorem_42_parallel_perpendicular_combination

theorem_43_midpoint_judgment

theorem_44_perpendicular_bisector_property_distance_equal

theorem_45_perpendicular_bisector_judgment

theorem_46_bisector_property_line_ratio

theorem_47_bisector_property_angle_equal

theorem_48_bisector_judgment_angle_equal

theorem_49_median_property

theorem_50_median_judgment

theorem_51_altitude_property

theorem_52_altitude_judgment

theorem_53_neutrality_property_similar

theorem_54_neutrality_property_angle_equal

theorem_55_neutrality_property_line_ratio

theorem_56_neutrality_judgment

theorem_57_circumcenter_property_line_equal

theorem_58_circumcenter_property_intersect

theorem_59_circumcenter_judgment

theorem_60_incenter_property_line_equal

theorem_61_incenter_property_intersect

theorem_62_incenter_property_judgment

theorem_63_centroid_property_line_equal

theorem_64_centroid_property_intersect

theorem_65_centroid_property_judgment

theorem_66_orthocenter_property_line_equal

theorem_67_orthocenter_property_intersect

theorem_68_orthocenter_property_judgment

theorem_69_congruent_property_line_equal

theorem_70_congruent_property_angle_equal

theorem_71_congruent_property_area_equal

theorem_72_congruent_judgment_sss

theorem_73_congruent_judgment_sas

theorem_74_congruent_judgment_asa

theorem_75_congruent_judgment_aas

theorem_76_congruent_judgment_hl

theorem_77_similar_property_angle_equal

theorem_78_similar_property_line_ratio

theorem_79_similar_property_perimeter_ratio

theorem_80_similar_property_area_square_ratio

theorem_81_similar_judgment_sss

theorem_82_similar_judgment_sas

theorem_83_similar_judgment_aa

theorem_84_triangle_perimeter_formula

theorem_85_triangle_area_formula_common

theorem_86_triangle_area_formula_heron

theorem_87_triangle_area_formula_sine

theorem_88_sine

theorem_89_cosine