

# Omega Prover

## **OmegaProver Documentation**

*Debug 0.2*

**OmegaProver Development Team**

September 28, 2022

# Omega Prover

**OmegaProver 文档**

*编辑版 0.2*

**OmegaProver 开发团队**

2022 年 9 月 28 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。

OmegaProver（之后简称 OP）几何问题求解系统可大致分为几部分：①OP 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。②OP 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。③OP 神经解析器：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 OP 形式化语言。④OP 神经预测器：为 OP 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

前言	1
一、OP 形式化系统	6
1.1 几何本体论	6
1.2 OP 形式化语言：语法及谓词设计	6
1.2.1 实体	6
1.2.2 属性	7
1.2.3 实体关系	7
1.2.4 代数式	7
1.2.5 定理	7
1.3 完备性理论	7
1.4 OP 形式化语言：图形的文字表示	7
二、OP 推理器	8
2.1 推理器设计	8
2.1.1 形式化语句解析与数据表示	8
2.2.2 定理实现和方程式求解	8
2.2 特色功能介绍	8
2.2.1 条件自动扩展和派生	8
2.2.2 推理过程展示与回溯	8
2.3 例题演示	8
三、OP 神经解析器	8
3.1 文本解析器	8
3.2 图像解析器	8
四、OP 神经预测器	9
4.1 OP 形式化语言 embedding	9
4.1.1 谓词 embedding	9
4.2.2 个体词 embedding	9
4.2 基于注意力机制的神经预测器	9
五、数据集标注指南	9
5.1 构图	9
图形 Shape(\$)	9
共线 Collinear(\$)	10
5.2 实体	10
点 Point(\$)	10
线 Line(\$)	11
角 Angle(\$)	11
三角形 Triangle(\$)	11
直角三角形 RightTriangle(\$)	12
等腰三角形 IsoscelesTriangle(\$)	12
等边三角形 EquilateralTriangle(\$)	12
多边形 Polygon(\$)	13
5.3 实体属性	13
线长度 Length(Line(\$))	13

角大小 Measure(Angle(\$)) .....	14
三角形面积 Area(Triangle(\$)) .....	14
三角形周长 Perimeter(Triangle(\$)) .....	14
三角形高 Altitude(Triangle(\$)) .....	15
5.4 实体关系 .....	15
中点 Midpoint(Point(\$),Line(\$)) .....	15
相交 Intersect(Point(\$),Line(\$),Line(\$)) .....	15
平行的 Parallel(Line(\$),Line(\$)) .....	16
无序平行 DisorderParallel(Line(\$),Line(\$)) .....	16
垂直的 Perpendicular(Point(\$),Line(\$),Line(\$)) .....	16
垂直平分线 PerpendicularBisector(Point(\$),Line(\$),Line(\$)) .....	17
角平分线 Bisector(Line(\$),Angle(\$)) .....	18
中线 Median(Line(\$),Triangle(\$)) .....	18
高 IsAltitude(Line(\$),Triangle(\$)) .....	18
中位线 Neutrality(Line(\$),Triangle(\$)) .....	19
外心 Circumcenter(Point(\$),Triangle(\$)) .....	19
内心 Incenter(Point(\$),Triangle(\$)) .....	19
重心 Centroid(Point(\$),Triangle(\$)) .....	19
垂心 Orthocenter(Point(\$),Triangle(\$)) .....	20
全等三角形 Congruent(Triangle(\$),Triangle(\$)) .....	20
相似三角形 Similar(Triangle(\$),Triangle(\$)) .....	20
镜像全等三角形 MirrorCongruent(Triangle(\$),Triangle(\$)) .....	21
镜像相似三角形 MirrorSimilar(Triangle(\$),Triangle(\$)) .....	21
5.5 表达式 .....	22
加 Add(expr1,expr2) .....	22
减 Sub(expr1,expr2) .....	22
乘 Mul(expr1,expr2) .....	22
除 Div(expr1,expr2) .....	22
幂 Pow(expr1,expr2) .....	23
正弦 Sin(expr) .....	23
余弦 Cos(expr) .....	23
正切 Tan(expr) .....	23
实数 R .....	23
表达式 expr .....	23
和 Sum(expr1,expr2, ...) .....	23
平均 Avg(expr1,expr2, ...) .....	23
相等 Equal(expr1,expr2) .....	24
5.6 解题目标 .....	24
Find(\$) .....	24
2.7 定理 .....	24
nous_1_area_addition .....	24
nous_2_line_addition .....	24
nous_3_angle_addition .....	24
nous_4_ .....	25

nous_5_.....	25
nous_6_.....	25
nous_7_.....	25
nous_8_.....	25
nous_9_.....	25
nous_10_.....	25
auxiliary_11_.....	25
auxiliary_12_.....	25
auxiliary_13_.....	25
auxiliary_14_.....	25
auxiliary_15_.....	25
auxiliary_16_.....	25
auxiliary_17_.....	25
auxiliary_18_.....	25
auxiliary_19_.....	25
auxiliary_20_.....	25
theorem_21_triangle_property_angle_sum .....	25
theorem_22_triangle_property_equal_line_angle .....	26
theorem_23_pythagorean.....	26
theorem_24_right_triangle_property_rt .....	26
theorem_25_right_triangle_property_special_rt .....	26
theorem_26_pythagorean_inverse .....	26
theorem_27_right_triangle_judgment.....	26
theorem_28_isosceles_triangle_property_angle_equal.....	26
theorem_29_isosceles_triangle_property_side_equal.....	26
theorem_30_isosceles_triangle_property_line_coincidence .....	26
theorem_31_isosceles_triangle_judgment_angle_equal.....	26
theorem_32_isosceles_triangle_judgment_side_equal.....	26
theorem_33_equilateral_triangle_property_angle_equal .....	26
theorem_34_equilateral_triangle_property_side_equal .....	26
theorem_35_equilateral_triangle_judgment_angle_equal .....	26
theorem_36_equilateral_triangle_judgment_side_equal .....	26
theorem_37_equilateral_triangle_judgment_isos_and_angle60 .....	26
theorem_38_intersect_property .....	26
theorem_39_parallel_property.....	26
theorem_40_parallel_judgment.....	27
theorem_41_perpendicular_judgment.....	27
theorem_42_parallel_perpendicular_combination .....	27
theorem_43_midpoint_judgment .....	27
theorem_44_perpendicular_bisector_property_distance_equal .....	27
theorem_45_perpendicular_bisector_judgment .....	27
theorem_46_bisector_property_line_ratio.....	27
theorem_47_bisector_property_angle_equal .....	27
theorem_48_bisector_judgment_angle_equal .....	27

theorem_49_median_property .....	27
theorem_50_median_judgment .....	27
theorem_51_altitude_property .....	27
theorem_52_altitude_judgment .....	27
theorem_53_neutrality_property_similar .....	27
theorem_54_neutrality_property_angle_equal .....	27
theorem_55_neutrality_property_line_ratio .....	27
theorem_56_neutrality_judgment.....	27
theorem_57_circumcenter_property_line_equal .....	27
theorem_58_circumcenter_property_intersect .....	28
theorem_59_circumcenter_judgment.....	28
theorem_60_incenter_property_line_equal.....	28
theorem_61_incenter_property_intersect.....	28
theorem_62_incenter_property_judgment .....	28
theorem_63_centroid_property_line_equal .....	28
theorem_64_centroid_property_intersect .....	28
theorem_65_centroid_property_judgment.....	28
theorem_66_orthocenter_property_line_equal .....	28
theorem_67_orthocenter_property_intersect .....	28
theorem_68_orthocenter_property_judgment.....	28
theorem_69_congruent_property_line_equal .....	28
theorem_70_congruent_property_angle_equal .....	28
theorem_71_congruent_property_area_equal.....	28
theorem_72_congruent_judgment_sss .....	28
theorem_73_congruent_judgment_sas .....	28
theorem_74_congruent_judgment_asa .....	28
theorem_75_congruent_judgment_aas .....	28
theorem_76_congruent_judgment_hl .....	29
theorem_77_similar_property_angle_equal .....	29
theorem_78_similar_property_line_ratio.....	29
theorem_79_similar_property_perimeter_ratio .....	29
theorem_80_similar_property_area_square_ratio .....	29
theorem_81_similar_judgment_sss.....	29
theorem_82_similar_judgment_sas .....	29
theorem_83_similar_judgment_aa .....	29
theorem_84_triangle_perimeter_formula .....	29
theorem_85_triangle_area_formula_common .....	29
theorem_86_triangle_area_formula_heron .....	29
theorem_87_triangle_area_formula_sine .....	29
theorem_88_sine .....	29
theorem_89_cosine.....	29



## 一、OP 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 OP 形式化语言的设计思路。

### 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 OP 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 OP 形式化语言：语法及谓词设计

#### 1.2.1 实体

### 1.2.2 属性

### 1.2.3 实体关系

### 1.2.4 代数式

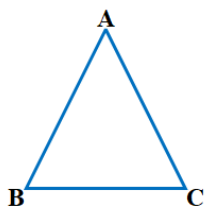
### 1.2.5 定理

## 1.3 完备性理论

## 1.4 OP 形式化语言：图形的文字表示

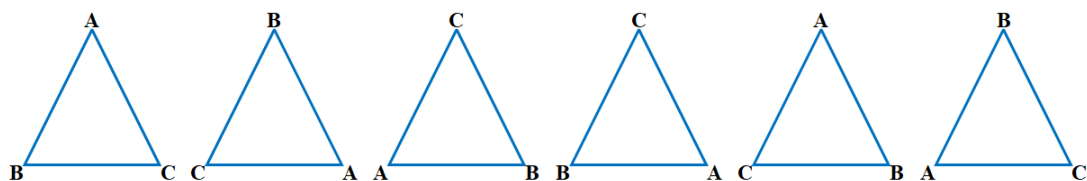
上一节介绍了完备性理论，我们指出一个正确的形式化系统应该建立正确的映射，不造成信息损失。几何问题的特点是既有文字描述，又有图像描述。在几何问题描述转化为 OP 形式化语言描述的过程中，文字信息是非常容易转化的，只需使其结构化即可，几乎不存在信息损失；对于图像信息的形式化，乍一想貌似需要考虑很多。接下来我们便分析如何将图形转化为形式化语言描述。

我们从 OP 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

如图 1.1.4 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

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 OP 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。

## 二、OP 推理器

### 2.1 推理器设计

#### 2.1.1 形式化语句解析与数据表示

#### 2.2.2 定理实现和方程式求解

### 2.2 特色功能介绍

#### 2.2.1 条件自动扩展和派生

#### 2.2.2 推理过程展示与回溯

### 2.3 例题演示

## 三、OP 神经解析器

### 3.1 文本解析器

### 3.2 图像解析器

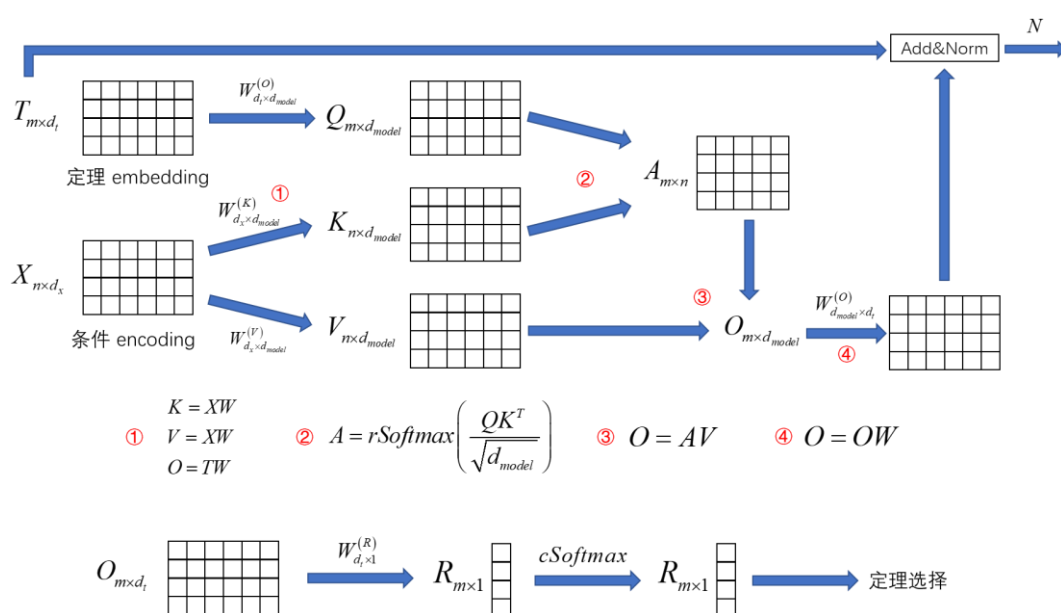
## 四、OP 神经预测器

### 4.1 OP 形式化语言 embedding

#### 4.1.1 谓词 embedding

#### 4.2.2 个体词 embedding

### 4.2 基于注意力机制的神经预测器



## 五、数据集标注指南

本章节对 OP 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 OP 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

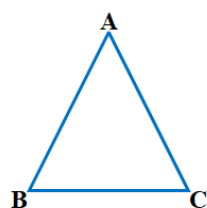
构图语句是最基本的形式化语句。推理器读取构图语句，自动扩展出所有的点、线、角和几何图形，实现 1.2 节中介绍的“拼图法”。

#### 图形 Shape(\$)

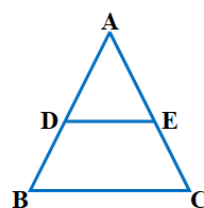
►描述：内部无连线、由直线组成的封闭图形定义为 shape。shape 由若干 point 组成，shape 的标注要求按照逆时针按顺序标出所有的点。根据选择的起始 point 不同，一个 shape 可以有多种标注形式，选择一种即可。

►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)}

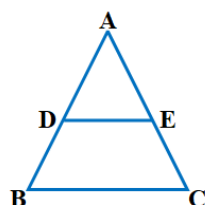
(b) **Shape(ADE), Shape(DBCE)**

### 共线 Collinear(\$)

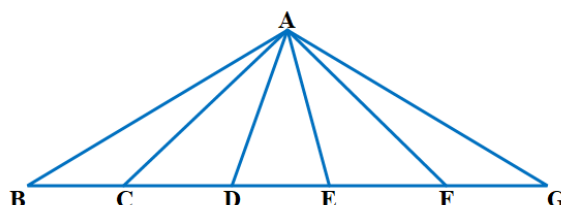
►描述：共线的若干点定义为 **collinear**。**collinear** 由若干 **point** 组成，**collinear** 的标注要求按照一条线上点的先后顺序标出所有的点。根据选择的起始 **point** 不同，一个 **collinear** 可以有 2 种标注形式，选择一种即可。

►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

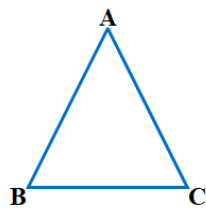
实体声明语句表示图形中的几何元素。

### 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)

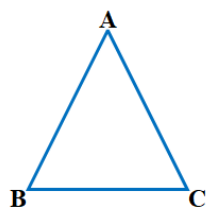
(a)  $\text{Point(A), Point(B), Point(C)}$

### 线 $\text{Line(\$)}$

►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)

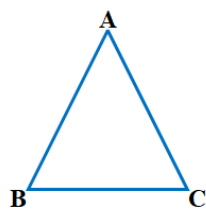
(a)  $\text{Line(AB), Line(BC), Line(AC)}$

### 角 $\text{Angle(\$)}$

►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)

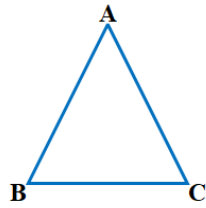
(a)  $\text{Angle(ABC), Angle(BCA), Angle(CAB)}$

### 三角形 $\text{Triangle(\$)}$

►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)

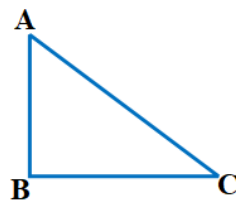
(a) **Triangle(ABC)**  
or {Triangle(BCA)  
Triangle(CAB)}

### 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)

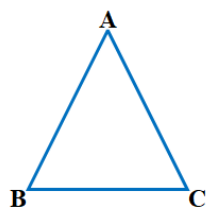
(a) **RightTriangle(ABC)**

### 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)

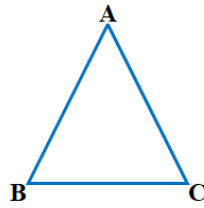
(a) **IsoscelesTriangle(ABC)**

### 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)

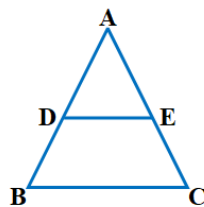
(a) **EquilateralTriangle(ABC)**  
 or {EquilateralTriangle(BCA)  
 EquilateralTriangle(CAB)}

### 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)



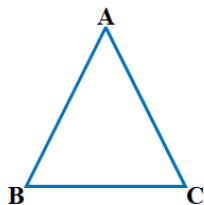
(a)  $\text{Length}(\text{Line}(\text{DE}))$   
or  $\{\text{Length}(\text{Line}(\text{ED}))\}$

### 角大小 $\text{Measure}(\text{Angle}(\$))$

►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)

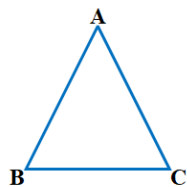
(a)  $\text{Measure}(\text{Angle}(\text{ABC}))$

### 三角形面积 $\text{Area}(\text{Triangle}(\$))$

►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)

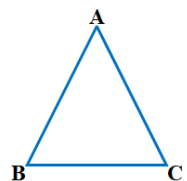
(a)  $\text{Area}(\text{Triangle}(\text{ABC}))$   
or  $\{\text{Area}(\text{Triangle}(\text{BCA}))$   
 $\text{Area}(\text{Triangle}(\text{CAB}))\}$

### 三角形周长 $\text{Perimeter}(\text{Triangle}(\$))$

►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))}

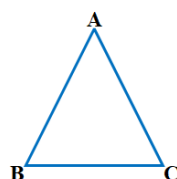
### 三角形高 Altitude(Triangle(\$))

►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)

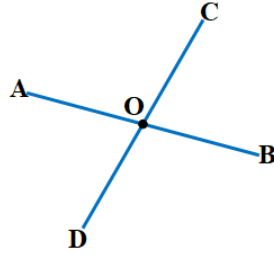
(a) **Midpoint(Point(M),Line(AB))**  
 or {Midpoint(Point(M),Line(BA))}

### 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)

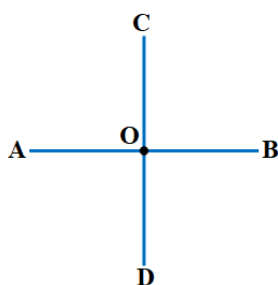
(a) `Parallel(Line(AB) ,Line(CD))`  
 or {`Parallel(Line(DC) ,Line(BA)),`  
`Parallel(Line(CD) ,Line(AB)),`  
`Parallel(Line(BA) ,Line(DC))`}

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

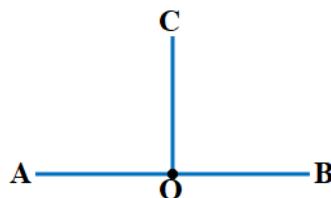
►描述：线与线的关系。与相交的标注方法一致。需注意的是，垂直的图形有 3 种变形，如例题所示。

►要点：①从左到右从上到下①4 种表示方式选其一

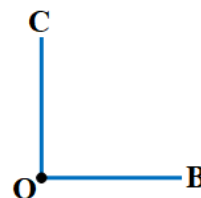
►标注示例：



5.4.4 (a)



5.4.4 (b)



5.4.4 (c)

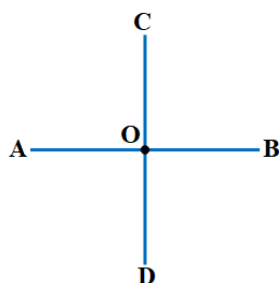
- (a) `Intersect(Point(O),Line(AB) ,Line(CD))`  
or {`Intersect(Point(O),Line(CD) ,Line(BA)),`  
`Intersect(Point(O),Line(BA) ,Line(DC)),`  
`Intersect(Point(O),Line(DC) ,Line(AB))`}}
- (b) `Intersect(Point(O),Line(AB) ,Line(CO))`  
or {`Intersect(Point(O),Line(CO) ,Line(BA)),`  
`Intersect(Point(O),Line(BA) ,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($))`

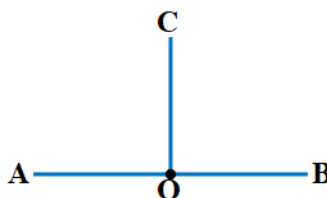
►描述：线与线的关系。与垂直的标注方法一致。第一条线表示被平分的线，第二条线表示垂直平分线，因此此关系只有 2 种表示方式。垂直的图形有 2 种变形，如例题所示。

►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))`

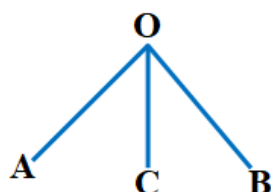
`Intersect(Point(O),Line(BA),Line(OC))}`

### 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)

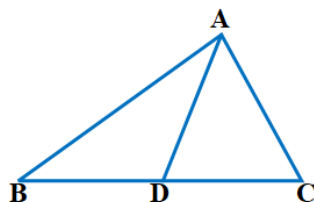
(a) `Bisector(Line(OC),Angle(BOA))`

### 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)

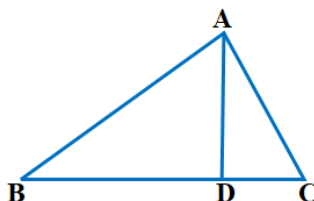
(a) `Median(Line(AD),Triangle(ABC))`

### 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)

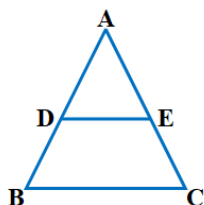
(a) `IsAltitude(Line(AD),Triangle(ABC))`

### 中位线 $\text{Neutrality}(\text{Line}(\$), \text{Triangle}(\$))$

►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)

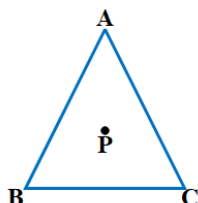
(a)  $\text{Neutrality}(\text{Line}(\text{DE}), \text{Triangle}(\text{ABC}))$

### 外心 $\text{Circumcenter}(\text{Point}(\$), \text{Triangle}(\$))$

►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)  $\text{Circumcenter}(\text{Point}(\text{P}), \text{Triangle}(\text{ABC}))$

or  $\{\text{Circumcenter}(\text{Point}(\text{P}), \text{Triangle}(\text{BCA})),$   
 $\text{Circumcenter}(\text{Point}(\text{P}), \text{Triangle}(\text{CAB}))\}$

### 内心 $\text{Incenter}(\text{Point}(\$), \text{Triangle}(\$))$

►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)  $\text{Incenter}(\text{Point}(\text{P}), \text{Triangle}(\text{ABC}))$

or  $\{\text{Incenter}(\text{Point}(\text{P}), \text{Triangle}(\text{BCA})),$   
 $\text{Incenter}(\text{Point}(\text{P}), \text{Triangle}(\text{CAB}))\}$

### 重心 $\text{Centroid}(\text{Point}(\$), \text{Triangle}(\$))$

►描述：三角形三边中线的交点。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：

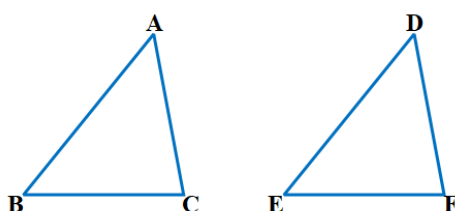
(a) **Orthocenter(Point(P),Triangle(ABC))**  
or {Orthocenter(Point(P),Triangle(BCA)),  
Orthocenter(Point(P),Triangle(CAB))}

**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)

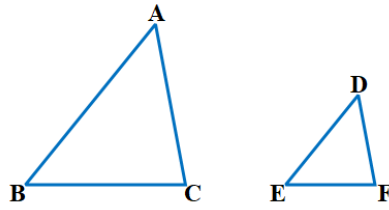
(a) **Congruent(Triangle(ABC),Triangle(DEF))**  
or {Congruent(Triangle(BCA),Triangle(EFD)),  
Congruent(Triangle(CAB),Triangle(FDE)),  
Congruent(Triangle(DEF),Triangle(ABC)),  
Congruent(Triangle(EFD),Triangle(BCA)),  
Congruent(Triangle(FDE),Triangle(CAB))}

**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)

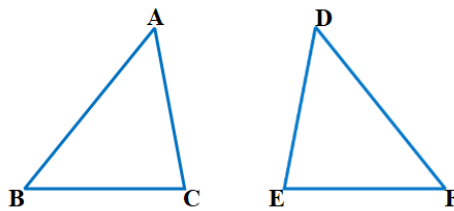
(a) **Similar(Triangle(ABC),Triangle(DEF))**  
 or {Similar(Triangle(BCA),Triangle(EFD)),  
 Similar(Triangle(CAB),Triangle(FDE)),  
 Similar(Triangle(DEF),Triangle(ABC)),  
 Similar(Triangle(EFD),Triangle(BCA)),  
 Similar(Triangle(FDE),Triangle(CAB))}

### 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， $\triangle ABC$  按照点一一对应得到的全等三角形应是 $\triangle DFE$ ；但 $\triangle DFE$  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， $\triangle ABC$  镜像全等 $\triangle DEF$ 。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))}

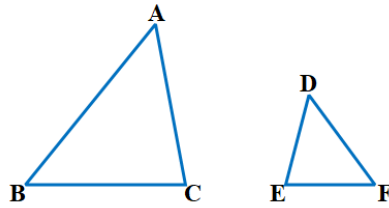
### 镜像相似三角形 MirrorSimilar(Triangle(\$),Triangle(\$))

►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， $\triangle ABC$  按照点一一对应得到的相似三角形应是 $\triangle DFE$ ；但 $\triangle DFE$  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， $\triangle ABC$  镜像相似 $\triangle DEF$ 。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：  $\text{expr1}+\text{expr2}$

▶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：  $\text{expr1}-\text{expr2}$

▶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：  $\text{expr1}*\text{expr2}$

▶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:  $\text{expr1}/\text{expr2}$

►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:  $\text{expr1}^{\text{expr2}}$

►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:  $\text{@expr}$

►示例: `Sin(x)`  
`Sin(60)`  
`Sin(Measure(Angle(ABC)))`

### 余弦 Cos(expr)

►对应表达式:  $\text{\#expr}$

►示例: `Cos(Add(x, y))`  
`Cos(60)`  
`Cos(Measure(Angle(ABC)))`

### 正切 Tan(expr)

►对应表达式:  $\text{\$expr}$

►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:  $\text{expr1}+\text{expr2}+\cdots$

►示例: `Sum(x+3, y+2, z+1)`

### 平均 Avg(expr1,expr2, ...)

▶对应表达式:  $\{expr1+expr2+\cdots+expr_n\}/n$

▶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