模型论域映射函数及其在模型组合中的作用

刘志模[1]　陈　振[2]　王晓山[2]　刘　徽[3]
[1](西南计算机公司，重庆市，400060), [2,](北京理工大学，北京市，100081)
[3](中国兵器工业计算机应用技术研究所，北京市，100089)

摘　要： 基于模型组合对模型论域的要求，本文将模型论域进一步分解为变量定义域和函数值域，论述了模型受限并运算的本质是在并运算中保持了函数和关系的定义域不变，以使并模型中所涉函数和关系成立有效。在论域分析的基础上，本文提出了论域映射函数方法，建立了模型之间数据的相互引用机制，提高了模型组合处理数据的灵活性和效能。与此同时，本文还论述了论域映射函数方法不会影响模型受限并运算的实现，分析了论域映射函数方法对多种模型组合稳定性的影响因素及解决办法。

关键词：　模型论域，变量定义域，函数值域，论域映射函数，数据循环引用
中图分类号： TP181　　　　**文献标志码：** A

Model domain mapping function and its role in model combinations

Liu Zhimo[1]　Chen Zhen[2]　Wang xiao shan[2]　Liu Hui[3]
[1](XiNan Computer Corp., ChongQing, 400060, China),
[2](Beijing Institue of Technology，Beijing，100081，China),
[3](China Ordnance Industry Computer Application Technology Research Institute, Beijing，100089，China)

Abstract: Based on the requirements of model combinations for the model domain, the model domain is decomposed into variable definition domain and fanction range in this paper. It has been confirmed that the essence of the limited parallel operation of models is to maintain the definition domain of functions and relationships unchanged during the parallel operation, in order to make the functions and relationships involved in the parallel model valid. On the basis of domain analysis, a method on the domain mapping function is proposed in the paper.　A mechanism for mutual quoting of the data between models is established. The flexibility and efficiency of processing data in model combinations are Improved. At the same time, the article also discusses that the domain mapping function method will not affect the implementation of the limited parallel operation of models, and the influencing factors and solutions on the stability of system are analyzed in various model combinations due to the introduction of domain mapping functions .

Keywords: model domain，variable definition domain，fanction range，domain mapping function，data circular reference

一. 引　言

当前，人工智能进入了飞速发展时期。Trangsformer、BERT、ChatGPT-3、ChatGPT-4等大模型相继诞生[1][2][3]，它们在自然语言和多模态信息处理方面取得了突出的进步，实现了文本、编程、图像、语音、视频、音乐等多模态信息的处理和转换。以此为基础的生成式人工智能技术取得了瞩目进展。现在，在 Trangsformer 基础上发展起来的预训练大模型已能调节超千亿参数进行深度学习，成为当下人工智能领域的主流方向。近年来，国内也涌现出以 DeepSeek 为代表的大模型应用，成绩瞩目，成为全球人工智能领域的焦点[4]。

但另一方面亦应看到，这些大模型通过调节内部参数来进行学习，模型具有高度复杂性和非线性，存在可解释性和复现困难的问题[5]。它们的模型原理是什么，能否直接用于其他智能体设备的研发，这些都是人们关心，但直接从当前的大模型获取知识又相当困难的问题。从外部看，它们具有人们需要的功能，但又未使用具体的数学公式去描述它，可重复性欠缺，只能将其看成为具有特定功能的黑箱[6][7]。人们期待对此有更深刻的理论和数学描述。随着多模态大语言模型，知识图谱，人工智能与多模态数学推理，具身智能，交叉领域应用等人工智能领域的技术研究飞速进步，若干基础理论研究工作有必要同步跟进，以支持人工智能领域技术的更快更好发展。

毋容置疑，数理逻辑理论是当下人工智能基础理论的重要组成部分。数理逻辑的模型论对模型的概念有严格的定义，并建立了一套严格的理论体系，可作为人工智能的基础理论进行研究。其中的模型概念是由形式语言通过解释（映射）所定义产生的，包含了若干函数和关系式，组成为一个完整的逻辑整体。由于这种模型通过形式语言严格进行了定义，满足形式语言的推理规则，由它产生的逻辑推理结论应该适用于一般的实体模型推理。因此，研究数理逻辑模型对人工智能系统的研发有着重要的现实意义。

就模型而言，人们期望的一种情景是：可以由若干简单模型组合成更大的模型；或反过来，复杂的较大模型可以分解为若干较小的简单模型来求解。这样，利用相对简单、成熟的模型能够组合产生复杂的、较大的模型。这对人工智能系统、独立智能体的研究来说有重要的意义。要实现模型的组合就必须要求进行模型的并运算，然而在数理逻辑模型论中，一般情况下的模型并运算是不满足运算封闭要求的[8]，因此无法定义，也就无法实现模型的并运算。现有模型论理论对此未能进行深入研究，这是一种缺失。我们在《一种树形模型链构建方法及其应用展望》[9] 一文中，提出了数理逻辑模型论中的模型受限并运算方法，实现了模型在一定约束条件下的并运算；论述了由若干小模型组合成大模型，或将大模型分解为小模型来实现的建模途径。这对人工智能设备的研究是一种理论支持。但模型受限并运算的成立是建立在并运算时对论域的强制限制来实现的，这对人工智能系统的构建又似乎显得有些灵活性不足，影响到系统运行的效率。因此，本文将从模型的论域角度对这一问题作进一步的探讨分析，以期改进模型组合的灵活性和运行效率问题。

二. 模型受限并运算回顾

为方便对利用数理逻辑理论对建模问题进行深入讨论，这里先将本文所涉及的数理逻

辑模型论及模型组合问题作一简要回顾。

1. 一阶语言符号系统

本文所涉一阶逻辑要素，包括：论域 M、个体记号集合 C、关系记号集合 R、函数记号集合 F、变元记号集合$\{x, y,\ldots\}$、逻辑运算符集合$\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$、量词记号集合$\{\forall, \exists\}$、标点符号集合{ (,) , , }组成，可参见一般数理逻辑书籍。

以上记号集合除论域外，其余集合组成了一阶形式语言$\mathcal{L}$：

$$\mathcal{L} = \{a, \ldots, R, \ldots, \approx, f, \ldots, x, \ldots, u, \ldots, \neg, \ldots, \forall, \exists, (\ ,\ )\ ,,\ \}\ 。$$

其中，$\{\approx, x, \ldots, u, \ldots, \neg, \ldots, \forall, \exists, (\ ,\ )\ ,,\ \}$ 是不同系统中的共有部分，一般在特定系统中就不再列出。

定义 2.1: 设

$C = \{c_1, c_2, \ldots, c_\kappa\}$ 是一些个体记号组成的集合，

$R = \{r_1, r_2, \ldots, r_\lambda\}$ 是一些关系记号组成的集合，

$F = \{f_1, f_2, \ldots, f_\mu\}$ 是一些函数记号组成的集合，

则 $\mathcal{L} = \{C, R, F\} = \{c_1, c_2, \ldots, c_\kappa, r_1, r_2, \ldots, r_\lambda, f_1, f_2, \ldots, f_\mu\}$构成一个特定的语言，其中，$C$、$R$、$F$ 的基数 κ、λ、μ 可以是有限或者无限基数。

本文中所使用的一阶逻辑的项、公式、语句等的定义和形成规则，以及变元在论域中的取值解释等同一般文献[8][10][11][12]。

2. 模型定义

数理逻辑模型是一种结构性模型，为讨论方便，现将有关定义详尽列出。

定义 2.2: 设$\mathcal{L}$为一阶语言。一个 $\mathcal{L}$-结构是一个二元组 $\mathcal{M} = (M, I)$，其中 M 为非空集合，称为 $\mathcal{M}$ 的论域，I 为定义在 $\mathcal{L}$ 上满足如下条件的映射(称 I 为**解释**)：

1） I 把 $\mathcal{L}$ 中的每个常项符 c 解释为 M 中的元素，即 $I(c) \in M$ ；
2） I 把 $\mathcal{L}$ 中的每个 n 元关系符 P 解释为 M 上的 n 元关系，即 $I(P) \subseteq M^n$ ；
3） I 把 $\mathcal{L}$ 中的每个 n 元函数符 f 解释为 M 上的 n 元函数，即 $I(f)$ 为从 M^n 到 M 的映射，$I(f)$：$M^n \rightarrow M$ 。

常把 $I(c)$、$I(P)$、$I(f)$ 分别记为 c^M、、P^M、f^M 。则对如上的 $\mathcal{L}$-结构，有 $c^M \in M$，$P^M \subseteq M^n$，f^M：$M^n \rightarrow M$。

在一些资料[8][12]中，将二元组 $\mathcal{M} = (M, I)$看成为模型是针对 $\mathcal{L}$ 中的公式 A 或公式集 Σ 而定义的，即当 $\mathcal{M} \vDash$ A(或 $\mathcal{M} \vDash \Sigma$)时，定义 $\mathcal{M}$ 是 A(或 Σ)的模型。而在其它资料[10]中，把二元组 (M, I)看成为对语言 $\mathcal{L} = \{C, R, F\}$在 M 上的一种解释，当该二元组已把 $\mathcal{L}$ 解释为$\{\bar{C}, \bar{R}, \bar{F}\}$后，则把 $\mathfrak{A} = (M, \bar{C}, \bar{R}, \bar{F})$ 叫做语言 $\mathcal{L}$ 的一个模型，M 叫做 $\mathcal{L}$ 的论域。

从结构上看，二者实际上是一致的。本文只是从数据结构上来探讨逻辑模型的问题，因此，采用[10]的方法，将把论域上实现解释后的结构看成为语言的模型。

定义 2.3: 设语言 $\mathcal{L} = \{C, R, F\}$。如果二元组$(M, I)$将 $\mathcal{L}$ 的个体、关系和函数记号在 M 中对应地作出了解释$\{\bar{C}, \bar{R}, \bar{F}\}$，则把 $\mathcal{M} = (M, \bar{C}, \bar{R}, \bar{F})$称作 $\mathcal{L}$ 的**模型**，M 称作 $\mathcal{M}$ 的

论域[10]。

本文以下讨论中，在提及语言 $\mathcal{L}=\{C, R, F\}$ 的模型时，已默认将 $\mathcal{L}$ 的个体、关系和函数记号在 M 中对应地作出了解释，为方便计，亦将 $\mathcal{M}=(M, C, R, F)$ 称作 $\mathcal{L}$ 的模型。

由以上数理逻辑模型的定义可看出，这里所涉的模型的结构性特征明显。而对一般实际应用中的模型而言，其针对的是现实世界中的具体对象，处理的是具体的实际问题，二者是不能等同的。数理逻辑的模型虽是一种结构模型，但它包含了抽象的数据对象、对象间的逻辑关系和推理规则、对象间的函数映射关系等等。同时，这些对象、关系和函数也构成了一个逻辑整体。而具体对象的模型一般也包含了处理的对象、数据、函数、推理方法等，只是所涉及的都是具体的对象实体，与数理逻辑的模型虽不能等同，但数理逻辑模型理论所得到的建模规则，推理规则等，可视为是具体模型应用中应当遵循的逻辑原则。因此，对具体模型的研究来说，研究数理逻辑模型有着重要的现实意义。

3. 子模型

定义 2.4: 设 $\mathcal{M}$，$\mathcal{N}$ 为同一语言 $\mathcal{L}=\{C, R, F\}$ 的两个模型。称 $\mathcal{M}$ 是 $\mathcal{N}$ 的子模型，记作 $\mathcal{M} \subseteq \mathcal{N}$，如果下列条件成立：

1） $M \subseteq N$ （M，N 分别为模型 $\mathcal{M}$，$\mathcal{N}$ 的论域）;

2） 对任意常项符 $c \in \mathcal{L}$，有 $c^M = c^N$；

3） 对任意 n-元关系符 $R \in \mathcal{L}$，及任意 $a_1, \ldots, a_n \in M$, 都有

$$R^M(a_1, \ldots, a_n) \text{ 当且仅当 } R^N(a_1, \ldots, a_n);$$

4） 对任意 n-元函数符 $f \in \mathcal{L}$，及任意 $a_1, \ldots, a_n \in M$, 都有

$$f^M(a_1, \ldots, a_n) = f^N(a_1, \ldots, a_n)。$$

对第 2）条，实际隐含 $c^M \in M$。

4. 模型受限并运算与模型组合

人们研究模型问题的一个主要目的就是为了实现模型组合。而为了实现模型组合，就离不开模型的串、并联，并由此实现小模型组合成更大更复杂模型的愿景。资料[9]对此已有详细论述。为本文讨论方便，现将有关定义汇集于后：

对于模型的交运算，有以下定义：

定义 2.5: 设模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 为一阶语言 $\mathcal{L}=\{C, R, F\}$ 的两个模型，其中 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$。定义 $\mathcal{M}_0 = \mathcal{M}_1 \cap \mathcal{M}_2 = (M_0, C_0, R_0, F_0)$ 为二者的交运算，满足下列条件：

1） 论域 $M_0 = M_1 \cap M_2$；

2） 对任意常项符 $c \in \mathcal{L}$，有 $c^{M_0} = c^{M_1}$ 且 c^{M_2}；

3） 对任意 n-元关系符 $R \in \mathcal{L}$，及任意 $a_1, \ldots, a_n \in M_0$, 都有 $R^{M_0}(a_1, \ldots, a_n)$ 当且仅当（$R^{M_1}(a_1, \ldots, a_n)$ 且 $R^{M_2}(a_1, \ldots, a_n)$）；

4） 对任意 n-元函数符 $f \in \mathcal{L}$，及任意 $a_1, \ldots, a_n \in M_0$, 都有 $f^{M_0}(a_1, \ldots, a_n) = (f^{M_1}(a_1, \ldots, a_n)$ 且 $f^{M_2}(a_1, \ldots, a_n))$。

资料[9]已说明，模型的交运算对常项符、关系和函数都是封闭的。

对于模型的并运算，一般情况下存在运算不封闭的情况。因此，资料[9]提出了模型受限并运算的方法，实现了在一定约束条件下的模型并运算。

定义 2.6:(**模型受限并运算**)设模型 $\mathcal{M}_1,\mathcal{M}_2$ 为一阶语言 $\mathcal{L}=\{C,R,F\}$的两个模型，其中$\mathcal{M}_1=(M_1,C_1,R_1,F_1)$，$\mathcal{M}_2=(M_2,C_2,R_2,F_2)$。定义 $\mathcal{M}_0=\mathcal{M}_1\cup_{\text{lim}}\mathcal{M}_2=(M_0,C_0,R_0,F_0)$为模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 的受限并运算，如果满足如下条件：

1) 论域 $M_0=M_1\cup M_2$；
2) 对任意常项符 $\mathrm{c}\in\boldsymbol{\mathcal{L}}$，如果 $\mathrm{c}^{M_1}=\mathrm{c}^{M_0}$，则 $\mathrm{c}\in C_1$ 且 $\mathrm{c}\in C_0$；如果$\mathrm{c}^{M_2}=\mathrm{c}^{M_0}$，则 $\mathrm{c}\in C_2$ 且 $\mathrm{c}\in C_0$；
3) 对任意 n-元关系符 $R\in\boldsymbol{\mathcal{L}}$，及任意 $a_1,\ldots,a_n$，如果 $a_1,\ldots,a_n\in M_1$，则有 $R^{M_1}(a_1,\ldots,a_n)$ 当且仅当 $R^{M_0}(a_1,\ldots,a_n)$；如果 $a_1,\ldots,a_n\in M_2$，则有 $R^{M_2}(a_1,\ldots,a_n)$ 当且仅当 $R^{M_0}(a_1,\ldots,a_n)$。
4) 对任意 n-元函数符 $f\in\boldsymbol{\mathcal{L}}$，及任意 $a_1,\ldots,a_n$，如果 $a_1,\ldots,a_n\in M_1$，则有 $\boldsymbol{f}^{M_1}(a_1,\ldots,a_n)=f^{M_0}(a_1,\ldots,a_n)$；如果 $a_1,\ldots,a_n\in M_2$，则有 $f^{M_2}(a_1,\ldots,a_n)=f^{M_0}(a_1,\ldots,a_n)$。
5) 对任意 n-元变量取值组合 $a_1,\ldots,a_i,\ldots,a_n\in M_0$，其取值范围：只存在 $(\forall a_i\in M_1)\vee(\forall a_i\in M_2)$的情形(包括 $a_i\in M_1\cap M_2$)。

以上条件称作**模型受限并运算条件**。

资料[9]说明了下述引理成立：

引理 2.1： 模型受限并运算对关系和函数是封闭的。

资料[9]还列出了子模型的定义、模型交运算方法等，同时定义了子模型并运算的方法。

定义 2.7: 设一阶语言 $\mathcal{L}=\{C,R,F\}$的模型 $\mathcal{M}_1=(M_1,C_1,R_1,F_1),\mathcal{M}_2=(M_2,C_2,R_2,F_2)$是模型 $\mathcal{M}_0=(M_0,C_0,R_0,F_0)$ 的子模型，$\mathcal{M}_1\subseteq\mathcal{M}_0$，$\mathcal{M}_2\subseteq\mathcal{M}_0$。定义 $\mathcal{M}_0'=(M_0',C_0',R_0',F_0')=\mathcal{M}_1\cup_{sub}\mathcal{M}_2$ 为子模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 的并模型，如果满足如下条件：

1) 论域 $M_1\cup M_2=M_0'\subseteq M_0$；
2) $\mathcal{M}_1$，$\mathcal{M}_2$ 的并运算满足定义 3.2 中的模型受限并运算条件。

资料[9]指出，定义 2.7 确定的子模型并运算 $\cup_{\text{sub}}$是由受限模型并运算 $\cup_{\text{lim}}$ 所限定的，而 $\cup_{\text{sub}}$ 更能体现出并运算结果的子模型特征。

三. 模型的论域分解

3.1 模型组合对论域的要求

如前所述，我们进行模型组合的目的，是为了实现由若干小模型组合成大模型，以便处理更复杂的问题。这种组合应当是有意义的，内部协调且无矛盾的；而不是拼凑的，为组合而组合的。模型组合的一个突出优点是模型之间可以相互利用模型处理数据的结果，即一些模型的数据处理结果被整个大的模型看成为中间结果而被其他模型加以利用。这对大的模型系统的开发而言，作用是显然的。从直观上看，一个模型要能够利用其它模型处

理数据的结果，则必需共享其它模型处理后的数据，实现模型间数据传递。从模型论的角度看，能够相互传递数据的模型之间的数据域一定存在交集，故以下引理成立：

引理3.1： 设 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$ 和 $\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$ 是一阶语言 $\mathcal{L}=\{C, R, F\}$ 的两个模型。如果要在 $\mathcal{M}_1$ 和 $\mathcal{M}_2$ 之间直接实现数据传递，其必要条件是它们的论域 M_1 和 M_2 之交集一定非空，即有 $M_1 \cap M_2 \neq \Phi$。

对于模型受限并运算来说，在并运算中对 n-元变量的取值范围作了限定，这似乎又限制了使用论域的范围，因此有必要对此作进一步的深入讨论。

3.2 论域分解

定义 3.1： 设 $\mathcal{M} = (M, C, R, F)$为一阶语言 $\mathcal{L} = \{C, R, F\}$ 的模型，M 为模型 $\mathcal{M}$ 的论域，则称 $M = M_C \cup M_D \cup M_R$ 为 M 的论域分解，其中：

M_C ：常量域（constant），含一般常量：a，b，… ；逻辑常量： T(True)，F(Faulse)；

M_D ：定义域（domain），为模型中变量的取值范围；

M_R ：值域（Range），为模型中函数输出的取值范围。

在定义 3.1 中，M_R 只对函数做了定义，因关系是一种逻辑表达式，只有成立与不成立两种情况，其取值可包含在逻辑常量 {T，F } 中。

模型的定义域和值域可做进一步分解。

定义 3.2： 模型定义域 M_D 可进一步分解为 $M_D = M_{DR} \cup M_{DF}$。其中：

M_{DR} ：关系定义域，为模型所有关系表达式中变量的取值范围；

M_{DF} ：函数定义域，为模型所有函数中变量的取值集合。

在一个模型中，关系变量 $(x_1, x_2, \ldots, x_n)$ 与函数变量 $(y_1, y_2, \ldots, y_m)$ 的取值范围有可能存在重迭，即存在 $M_{DR} \cap M_{DF} \neq \Phi$。

一般情况下，一个模型可能包含不止一个函数，而是一个函数集合 $F = \{f_i\}(1 \leq i \leq n)$，函数 f_i 的定义域可表示为 M_{Df_i}，则依据定义 2.2，对函数的定义域有以下引理：

引理3.2： 设模型 $\mathcal{M}$ 的函数集为 $F = \{f_i\}(1 \leq \mathrm{i} \leq \mathrm{n})$，函数 f_i 的定义域为 M_{Df_i}，则模型的函数定义域为 $M_{DF} = \cup M_{Df_i}$。

定义 3.3： 函数 f 的取值范围称作 f 的值域，记作 M_{Rf}。

依据定义 3.3，有以下引理：

引理3.3： 设模型 $\mathcal{M}$ 的函数集为 $F = \{f_i\}(1 \leq i \leq n)$，则模型$\mathcal{M}$ 的函数值域为 $M_{RF} = \cup M_{Rf_i}$，且模型值域 $M_R = M_{RF}$。

在数理逻辑中，函数的定义域和值域均是一种结构性定义，与具体对象不同，它们具有更宽范的含义。比如，对函数定义域而言，它具有以下内涵：

从数据属性来看，定义域中的数据可以是普通数据，也可以是对象数据，…… ；

从数据类型来看，定义域中的数据可以是实数，虚数，整数，…… ；

从数据结构来看，定义域中的数据可以是单个数据，向量，矩阵，…… 。

对一个具体的模型来说，其定义域中的数据属性、数据类型、数据结构都是确定的，

而对于一般模型而言，其定义域中的数据属性、数据类型、数据结构可以不被指定，我们只是从结构上来研究定义域的特性。

3.3 函数封闭对定义域，值域的要求

将模型的论域按常数域、定义域、值域细分后，对一般的模型并运算造成的函数不封闭问题可以得到更深刻的认识。

资料[9]对模型并运算造成的函数不封闭问题有如下描述：

设模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 为一阶语言 $\mathcal{L}$ ={ C,R,F }的两个模型，其中 $\mathcal{M}_1=(M_1,C_1,R_1,F_1)$，$\mathcal{M}_2=(M_2,C_2,R_2,F_2)$。如果定义 $\mathcal{M}_0=\mathcal{M}_1\cup\mathcal{M}_2=(M_0,C_0,R_0,F_0)$，则首先要求 $M_0=M_1\cup M_2$，对某一 n-元函数 $f(x_1,\dots,x_n)\in\mathcal{L}$，它在 $\mathcal{M}_1$ 和 $\mathcal{M}_2$ 中都得到解释，对取值组合 $\{(a_1,\dots,a_n)|\ a_i\in M_0\}$，有以下几种情况：

1) $\{(a_1,\dots,a_n)|\ a_i\in M_0\ ,\ \forall a_i\in M_1\}$，这时 $f^{M_1}(a_1,\dots,a_n)\in M_1\ \Rightarrow$ $f^{M_0}(a_1,\dots,a_n)\in M_0$，运算封闭；
2) $\{(a_1,\dots,a_n)|\ a_i\in M_0\ ,\ \forall a_i\in M_2\}$，这时 $f^{M_2}(a_1,\dots,a_n)\in M_2\ \Rightarrow$ $f^{M_0}(a_1,\dots,a_n)\in M_0$，运算封闭；
3) 存在$\{\left(a_1,\dots,a_i,\dots,a_j,\dots,a_n\right)\middle|\ a_i,a_j\in M_0\ ,\ \exists\,(a_i\in M_1$ 且 $a_j\in M_2$且 $a_j\notin M_1\cap M_2)\}$，在这种情况下，$f^{M_0}\left(a_1,\dots,a_i,\dots,a_j,\dots,a_n\right)$的值无法确定是否在 $M_0=M_1\cup M_2$ 中，造成函数不封闭。

由于存在函数不封闭的情况，一般就无法定义模型的并运算。

从函数定义域的角度看，上述第 1）、2）种情况中，函数 $f^{M_1}(a_1,\dots,a_n)$和 $f^{M_2}(a_1,\dots,a_n)$自变量的取值均在函数的定义域内，故两个模型的函数和并模型的函数在此范围内是有定义的。而在第 3）种情况中，函数 $f^{M_0}\left(a_1,\dots,a_i,\dots,a_j,\dots,a_n\right)$的自变量取值已超出了已知函数 $f^{M_1}(a_1,\dots,a_n)$和 $f^{M_2}(a_1,\dots,a_n)$的定义域范围，故此造成了无法确定函数是否成立的问题，也就是造成了函数不封闭。

在模型并运算中造成函数不封闭的情况可以推广到模型的关系运算中。因为模型中的关系式中仍存在变量的定义域问题，故一般的模型并运算亦可能造成超出关系变量的定义域范围，进而造成关系表达式无法确定其是否成立的问题。

由以上分析可看出，以下定理成立。

定理3.1： 模型受限并运算方法的本质是在并运算中维持函数和关系的定义域不变，以保持并运算中的函数和关系成立。

定理 3.1 明确表示了模型受限并运算的本质和重要意义。从定理 3.1 的角度看，模型受限并运算亦可称为保定义域模型并运算。

3.4 模型定义域与值域的扩展

1. 单个模型定义域与值域的扩展

在数理逻辑模型论中，经常会遇到极大化问题。那么，在模型论域中，是否存在极大

化的可能呢？在模型论域细分后，这一问题就是显而易见的了。对任一单个模型 $\mathcal{M}$，其论域 $M = M_C \cup M_D \cup M_R$，假设对论域中一成员 $m \in M_R$，且 $m \notin M_D$，如果令 $M'_D = M_D \cup \{m\}$，最终有 $M_R \subseteq M'_D$。这是一种随意扩大模型定义域的行为，一般情况下是不成立的。如果需要扩大模型定义域，必须对扩大的成员从模型函数和关系式进行验证才行。

在实际应用中，一个成熟模型的使用常常会遇到功能扩展问题。一种是在新的数据环境中，需要扩大模型的函数功能，使之能够处理新的数据，提升模型的处理能力；二是适应新的模型环境，与其它模型协同工作，共同完成模型承担的任务。这两者都可能涉及到模型的修改问题。通过修改模型的实现功能，适应函数和关系定义域的变动，以及函数值域的变动，其结果是实现了模型功能和性能的提升。为了确认这些功能变化，最直接的办法就是对模型的使用进行验证。如果经过验证，这些改动是可行的，则模型的改动便是成功的；否则，模型的改动就是不可行的。

2. 模型组合中的论域扩展

在单个模型中，对模型的论域变动应当慎重处理，一般不能随意扩大模型的论域，其中特别不能随意扩大模型的定义域。但是，对模型组合而言，情况则有所不同。对通过模型受限并运算实现的并模型来说，其论域至少应当包含模型组合中各子模型的论域，论域规模自然是扩大了。

设模型组合 $\mathcal{M}_0 = (M_0, C_0, R_0, F_0)$ 是由若干子模型组合而成的并模型：$\mathcal{M}_0 = (\cup_{sub})_{i=1}^{n} \mathcal{M}_i = (\cup_{sub})_{i=1}^{n} (M_i, C_i, R_i, F_i)$，即模型组合 $\mathcal{M}_0$ 是由各子模型 $\mathcal{M}_i$(i =1 ,..., n) 通过模型受限并运算组合而成的模型，其中，$\mathcal{M}_0$ 的论域 $M_0 = \cup M_i$。显然，组合模型 $\mathcal{M}_0$ 的论域与各子模型相比是扩大了。但由于这里的模型并运算是受限并运算，由定理 3.1，在组合模型 $\mathcal{M}_0$ 中的各个子模型的函数和关系在运算中都不会超出它们的定义域范围，因此，组合模型在其论域 $M_0 = \cup M_i$ 的范围内是成立的。

对于不是由模型受限并运算组成的并模型的情形，由于一般情况下并运算不一定成立，因而使得并模型不一定成立。其原因是并运算时随着论域的扩大，子模型的定义域超出了原来确定的范围，造成并运算中的函数和关系不一定成立。因此，对于实际需要采用非受限并运算的并模型的情形，则需通过验证方式来确认并模型是否成立以及进一步的模型修改问题。

四. 论域映射函数及其在模型组合中的作用

4.1 论域映射函数

如前所述，我们进行模型组合的目的就是为了提升模型处理问题的能力，由若干小的成熟模型组合成更大型更复杂的模型，以便处理更复杂的实际问题。当然，这些组合在一起的模型应该构成一个有机整体，相互协同工作，充分发挥模型组合的整体性能。而要做到这一点，模型之间必须要能够实现处理后的数据相互共享和利用，即一些模型可以利用其他模型处理的数据结果，以提升自身处理问题的能力。在这种相互作用的机制下，使得

整个模型组合的整体性能有所提升，这正是我们进行模型组合所期望的。

资料[9]提出了模型受限并运算的方法，实现了模型的组合。如上所述，模型受限并运算的本质是在模型并运算时保持了函数和关系的定义域不变，这似乎又限制了模型受限并运算的运用范围。在实际应用中，为了解决在一般情况下因模型并运算的定义域受限导致其应用范围受限的问题，有两种解决办法：一种办法是对纯软件系统，可以将需要的子模型模块加入到模型组合中任何需要的地方来实现模型组合；而对于同时有硬件和软件构成的系统，比如具身智能系统，一般不太可能通过“复制”的方式来组合加载一个子模型模块，而是希望在需要的地方通过引用该模型模块处理后的数据来实现功能性组合。特别是，在由大量子模型组合成的复杂模型网络中，用复制子模型方式来得到它的数据处理结果是不可取的，也是行不通的。鉴于此，本文提出了一种使用论域映射函数的方法，用来实现一个模型对其它模型所处理数据的引用。

定义 4.1： 设模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 为一阶语言$\boldsymbol{\mathcal{L}} = \{C, R, F\}$ 的两个模型，其中 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$；设 $M_{1k} \subseteq M_1$为 M_1 的子论域，$M_{2p} \subseteq M_2$为 M_2 的子论域。称函数集 $F_{M_2-M_1}$ 为论域 M_2 到 M_1 的论域映射函数，如果 $F_{M_2-M_1}$ 将 M_{2p} 映射为 M_{1k}，即有 $F_{M_2-M_1}$： $M_{2p} \rightarrow M_{1k}$。

由定义 4.1 可看出，论域映射函数将其它模型的一部分论域映射到本模型的论域中，这为本模型直接引用其它模型的数据处理结果提供了可能。论域映射实质上是一种数据对应或变换，将一些数据形式变换成另外一些数据形式。当然这种变换在实际应用中应当是有确定含义的，且是可行的。定义 4.1 中未特别强调论域映射是值域或定义域之间的映射，是考虑到在将值域映射到另一模型的定义域时，也允许将一部分定义域同时映射的情况。一般情况下，定义 4.1 中的 M_{2p} 主要是指模型 $\mathcal{M}_2$ 的某些函数的值域，M_{1k} 主要是指模型 $\mathcal{M}_1$ 中的函数和关系的定义域。

例如，在一个软件中，对其它某个软件模块的调用，可视为将被调用模块输出的数据映射为调用模块的定义域中的数据，而由调用模块继续使用，这时的函数映射实质上就可用模块调用方式实现。又如，如果一个系统中既包含了二值逻辑系统，又包含有模糊逻辑系统，那么，模糊逻辑系统的处理结果可以被二值逻辑系统所采用。这时，模糊逻辑系统输出的数据需经过适当变换处理后再传递给二值系统，这时的函数映射就包括对模糊逻辑系统输出数据的引用和适当的数据变换[13]：模糊逻辑系统变量 x_A 的值域映射为二值逻辑系统 x_B的定义域时，需同时把 x_A 对应的隶属度 μ_{x_A} 从取值 [0,1] 变换为 x_B 的输入取值{0,1}，其变换可为

$$x_{B-In} = \begin{cases} 1 & \mu_{x_A} \geq 0.5 \\ 0 & \mu_{x_A} < 0.5 \end{cases}$$

由于在模型组合中引入了论域映射函数，有必要对模型的定义作适当修改，以体现论域映射函数在模型中的作用和定位。

定义 4.2：（包含论域映射函数的模型定义） 设一阶语言 $\boldsymbol{\mathcal{L}}=\{C, R, F\}$的一个模型为 $\mathcal{M} = (M, C, R, F)$。称 $\mathcal{M}' = (M, C, R, F \cup F_{Mo-M})$ 为在 $\mathcal{M}$ 中纳入了论域映射函数后的模型。其中，论域映射函数 F_{Mo-M} 为满足定义 4.1 的映射函数集，它的作用只是将其它模型的一

部分论域映射到模型 $\mathcal{M}$ 的论域 M 中，并不参与模型 $\mathcal{M}$ 内部的其它数据处理。

为方便计，之后我们提及包含论域映射函数的模型时，就直接采用 $\mathcal{M} = (M, C, R, F \cup F_{Mo-M})$表示了，其中在模型的函数集合中增加了论域映射函数 F_{Mo-M}，以区别于一般的模型。而对一般未采用论域映射函数的模型，仍采用 $\mathcal{M} = (M, C, R, F)$ 的形式表示。

模型中增加论域映射函数后，本模型的函数和关系可以引用其它模型映射到本模型定义域的数据。当然，在引用其它模型映射到本模型定义域中的数据时，使用者应该清楚这些数据的含义是与被引用模型的一种数据对应，而加以利用的。这样，对其它模型输出数据的引用，使得模型处理的数据范围变相扩大了，有可能获得整体性能的提升。

与此同时，为了获得性能的提升，也可能会涉及对关系集和函数集的部分修改。对于这一模型中关系和函数的修改问题，会涉及具体的模型组成，已超出本文讨论的范围，留待之后进一步研究了。

4.2 论域映射函数对模型受限并运算的影响

在模型中引入论域映射函数后，人们首先想到的问题是，模型受限并运算在添加论域映射函数后是否仍然成立？下述定理对这个问题给出了肯定的回答。

定理4.1： 模型中引入论域映射函数后，如果模型中的关系和函数未作调整改变，则以该模型为基础的模型受限并运算仍成立有效。

证明： 设 $\mathcal{M}_1$，$\mathcal{M}_2$ 为一阶语言 $\mathcal{L} = \{C, R, F\}$ 的两个模型，其中 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$。设 $\mathcal{M}_0 = \mathcal{M}_1 \cup_{\lim} \mathcal{M}_2 = (M_0, C_0, R_0, F_0)$ 为 $\mathcal{M}_1$，$\mathcal{M}_2$ 满足定义 2.2 条件的模型受限并运算结果。根据引理 2.1，以 $\mathcal{M}_1$，$\mathcal{M}_2$ 为基础的模型受限并运算对关系和函数是封闭的，即$\mathcal{M}_0$ 中的关系和函数 R_0, F_0 成立，在并运算后有确定的值。

现在 $\mathcal{M}_1$ 中引入论域映射函数 $F_{M_2-M_1}$: $M_{2r} \rightarrow M_{1q}$，其中，$M_{1q} \subseteq M_1$ 为 M_1的一个子论域，$M_{2r} \subseteq M_2$ 为 M_2 的一个子论域，$F_{M_2-M_1}$ 将模型 $\mathcal{M}_2$ 的子论域 M_{2r} 映射为模型 $\mathcal{M}_1$ 的子论域 M_{1q}。这时，模型 $\mathcal{M}_1$ 变成 $\mathcal{M}_1' = (M_1, C_1, R_1, F_1 \cup F_{M_2-M_1})$，$\mathcal{M}_2$ 仍保持不变。下面来考察模型 $\mathcal{M}_1'$ 与 $\mathcal{M}_2$ 的模型受限并运算 $\mathcal{M}_1' \cup_{\lim} \mathcal{M}_2 = \mathcal{M}_0' = (M_0', C_0', R_0', F_0')$。由于只在模型 $\mathcal{M}_1$ 中引入了论域映射函数 $F_{M_2-M_1}$，且只将 M_2 的子论域 M_{2r} 映射为 M_1的子论域 M_{1q}，与 $\mathcal{M}_1$ 相比，$\mathcal{M}_1'$ 的其它部分均未作改变，因此可以断言，$M_0' = M_0$，$C_0' = C_0$，$R_0' = R_0$，$F_0' = F_0 \cup F_{M_2-M_1}$，即有 $\mathcal{M}_0' = (M_0', C_0', R_0', F_0') = (M_0, C_0, R_0, F_0') = \left(M_0, C_0, R_0, F_0 \cup F_{M_2-M_1}\right)$。于是，对并模型 $\mathcal{M}_0'$ 而言，M_0 未改变；R_0 与 F_0 是封闭的，有确定的值，亦未改变；$F_{M_2-M_1}$有确定含义，从而并运算 $\mathcal{M}_0' = \mathcal{M}_1' \cup_{\lim} \mathcal{M}_2$ 是成立有效的。■

依据定理 4.1，在模型组合中引入了论域映射函数后，模型的受限并运算是函数封闭的，从并运算角度看，可以放心使用。但在模型组合中，如果不止一处采用了论域映射函数，方案是否依然可行？依据定理 4.1，可以肯定的是，模型受限并运算照样可以进行。但模型组合是一个整体，不止一处引入论域映射函数后，有可能对模型组合的整体性能造成影响，因此有必要对这一情况进行深入分析。

4.3 模型定义域和值域对论域映射函数的限制和影响

一般而言，论域映射函数是将其它模型的值域数据映射成本模型的定义域数据，以便本模型采用其它模型处理数据的成果。就两个相关模型的论域数据关联程度而言，存在论域重叠和不重叠两种情况。设有模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$ 和 $\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$。现在 $\mathcal{M}_1$ 中引入论域映射函数 $F_{M_2-M_1}$，使模型Ⅰ变成 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1 \cup F_{M_2-M_1})$，其中 $F_{M_2-M_1}$： $M_{2p} \rightarrow M_{1k}$。针对 M_{2p} 与 M_{1k} 是否存在重叠，分别讨论于后。

1. $M_{2p} \cap M_{1k} \neq \Phi$，二者存在重叠。

又分两种情况进行讨论。

a) $M_{2p} \subseteq M_{1k}$

对任意 $a \in M_{2p}$，都有 $b \in M_{1k}$，使得 $a = b$，于是 $a \in M_{1k}$，这时 $\mathcal{M}_1$ 可直接引用 M_{2p} 中的数据。以下定理成立：

定理4.2： 设有模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1 \cup F_{M_2-M_1})$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$，其中 $\mathcal{M}_1$ 包含论域映射函数 $F_{M_2-M_1}$: $M_{2p} \rightarrow M_{1k}$，且有 $M_{2p} \subseteq M_{1k}$，则 $\mathcal{M}_1$ 中通过论域映射函数引用 $\mathcal{M}_2$ 中的数据可以直接使用。

这是一种最理想最简洁的情况。

b) $M_{2p} \supset M_{1k}$

存在 $a \in M_{2p}$，使得 $a \notin M_{1k}$。这种情况下，在 $\mathcal{M}_1$ 中通过论域映射函数引用 $\mathcal{M}_2$ 中得到的值域数据时存在超出 $\mathcal{M}_1$ 定义域的情况，如果直接使用，则会出现函数不封闭和关系不成立。因此，为了使引用的数据在 $\mathcal{M}_1$ 中能正常使用，必须将 M_{2p} 的数据进行压缩。现通过一个数据压缩函数 F_{comp}将 M_{2p} 压缩为不超出 M_{1k}的范围来实现，即 F_{comp}: $M_{2p} \rightarrow M_{2p}'$，使得 $M_{2p}' \subseteq M_{1k}$ 成立。然后再通过论域映射函数 $F_{M_2-M_1}$: $M_{2p}' \rightarrow M_{1k}$ 在 $\mathcal{M}_1$ 中引用 $\mathcal{M}_2$ 得到的处理数据，根据定理 4.1，这是可行的。由此可得以下定理：

定理4.3： 设有模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1 \cup F_{M_2-M_1})$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$，其中 $\mathcal{M}_1$ 包含论域映射函数 $F_{M_2-M_1}$: $M_{2p} \rightarrow M_{1k}$，且有 $M_{2p} \supset M_{1k}$。这种情况下的论域映射可分两步实现：首先采用数据压缩函数 F_{comp} 压缩论域，F_{comp}: $M_{2p} \rightarrow M_{2p}'$，使得 $M_{2p}' \subseteq M_{1k}$ ；之后再用论域映射函数实现论域映射 $F_{M_2-M_1}'$: $M_{2p}' \rightarrow M_{1k}$。

定理 4.3 中的论域映射函数可看成为一个复合函数：

$$F_{M_2-M_1}: M_{2p} \rightarrow M_{1k} = F_{M_2-M_1}': (F_{comp}: M_{2p} \rightarrow M_{2p}') \rightarrow M_{1k}$$

由于在映射函数中采用了数据压缩，不可避免地会丢失被引用模型得到的一部分数据信息，这在使用中应当充分注意。

2. $M_{2p} \cap M_{1k} = \Phi$

在这种情况下，两个模型的论域数据无交集，一个模型无法直接引用另一个模型处理得到的数据成果。那么，能否间接利用另一个模型得到的处理结果呢？这是一个值得探索的问题。本文这里提出一个采用“数据代理”的方法，以期实现论域无交集条件下的论域映射。

定义 4.3： （论域数据代理）设一阶语言 $\mathcal{L}=\{C, R, F\}$ 的两个模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$，其中 M_{1k}，M_{2p}分别是两个模型论域的子论域：$M_{1k} \subseteq M_1$，

$M_{2p} \subseteq M_2$。在 $M_{2p} \cap M_{1k} = \Phi$ 的条件下，如果对于 M_{2p}中的某个成员 $b_i \in M_{2p}$，我们能够在 M_{1k} 中指定一个成员 $a_i \in M_{1k}$与之对应，且这种指定对在 $\mathcal{M}_1$ 中引用 $\mathcal{M}_2$ 的数据处理结果是有意义的，则称 a_i 为论域 M_2 的数据成员 b_i 在 M_1中的论域数据代理。

采用论域数据代理的方法，可以在两个模型的论域无交集的条件下实现模型之间的数据引用。当然，正如定义 4.3 所明确的，数据代理的指定是针对被引用数据而确定的，是在代理所属模型中数据处理时有明确针对被引用数据的含义，是有特别含义的，也就是在引用模型中有必要利用被引用模型处理的数据成果时才有意义。

有如一个模型中论域包含由张三、李四、王五、…… 组成的团队，我们可以在另外一个模型的论域中，采用 1、2、3、……编号与之一一对应，在采用编号的模型论域中，编号数字隐含着对应于原模型的团队成员。这里只是一个比喻说明。

模型的论域数据无交集而又要相互引用处理数据的情况应该是比较少的。大多数情况下，模型之间的处理数据要能实现相互利用，则在论域中应当存在数据重叠，即是 1 中所列出的情况。

4.4 论域映射函数对模型组合稳定性的影响

论域映射函数为模型组合中相互引用处理数据带来许多灵活性和方便，但在采用论域映射函数后，模型组合网络工作是否稳定？这是模型组合中引入论域映射函数后必须要回答的问题。在资料[9]中，模型组合有模型链、模型树、模型网络几种构成形式。现分别进行分析。

1. 论域映射函数对模型链的影响

资料[9]介绍了模型链的组成情况。设语言 $\mathcal{L} = \{\, C, R, F \,\}$的一系列模型为 $\mathcal{M}_\Gamma$： $\mathcal{M}_\gamma$，$\mathcal{M}_{\gamma-1}$，… ，$\mathcal{M}_1$，$\mathcal{M}_0$ ，其中 $\mathcal{M}_{i-1}$ 是 $\mathcal{M}_i$ 的子模型，即有 $\mathcal{M}_{i-1} \subseteq \mathcal{M}_i$（$\gamma \ge i \ge 1$， $\gamma \in \mathbf{I}$）。这一系列模型构成子模型升链：

$$\mathcal{M}_0 \subseteq \mathcal{M}_1 \subseteq \cdots \subseteq \mathcal{M}_\gamma \tag{4.1}$$

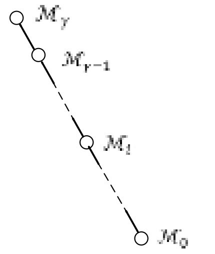


图 4.1　　子模型链结构

图 4.1 表示了式(4.1)模型链的结构。模型 $\mathcal{M}_0$ 处在模型链的最低层，模型 $\mathcal{M}_\gamma$ 处在模型链的最高层，低层的模型是高层模型的子模型。

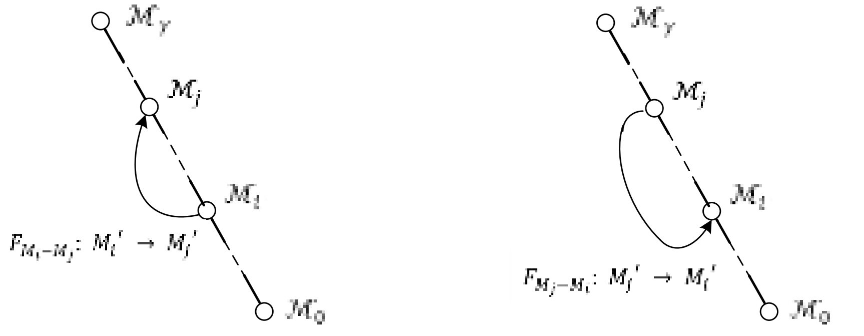
如果模型链构成一个数据处理整体，则从数据传递方向看，低层模型的输出数据是高一层模型的输入，逐级传递，且这种数据传递是单方向的。

现在来考察在子模型链中加入论域映射函数的情况。

设 $\mathcal{M}_i$，$\mathcal{M}_j$ 为图 4.1 模型链中的两个模型， $\mathcal{M}_i \subseteq \mathcal{M}_j$ （$i < j$）。现分别考察从 $\mathcal{M}_i$，$\mathcal{M}_j$ 处加入论域映射函数：

a)　在高端模型 $\mathcal{M}_j$ 中加入论域映射函数，引用低端模型 $\mathcal{M}_i$ 中的论域数据

两个模型可分别表示为 $\mathcal{M}_i = (M_i, C_i, R_i, F_i)$， $\mathcal{M}_j = (M_j, C_j, R_j, F_j \cup F_{M_i-M_j})$。其中 $F_{M_i-M_j}$: $M_i{}' \to M_j{}'$，$M_i{}' \subseteq M_i$，$M_i{}' \subseteq M_j$，如图 4.2 a) 所示。由于模型间存在 $\mathcal{M}_i \subseteq \mathcal{M}_j$，模型链中低端模型的数据效果一定会传递到高端模型中，因此，



a)　高端模型引用低端模型数据　　b)　低端模型引用高端模型数据

图 4.2　模型链中的论域映射

在模型链中再通过论域映射函数把低端模型的论域数据引用到高端模型中去，只会在量上和提前时间上影响到高端模型，而不会影响到模型链的稳定性。故有以下引理：

引理4.1：　在模型链中，在高端模型中通过论域映射函数引用低端模型的论域数据，不会影响模型链的稳定性。

b)　在低端模型 $\mathcal{M}_i$ 中加入论域映射函数，引用高端模型 $\mathcal{M}_j$ 中的论域数据

如图 4.2 b) 所示，在低端模型 $\mathcal{M}_i = (M_i, C_i, R_i, F_i \cup F_{M_j-M_i})$ 中加入论域映射函数，引入高端模型 $\mathcal{M}_j = (M_j, C_j, R_j, F_j)$的论域数据：$F_{M_j-M_i}$: $M_j{}' \to M_i{}'$，$M_i{}' \subseteq M_i$，$M_i{}' \subseteq M_j$ 。

定义 4.4： 设有子模型升链 $\mathcal{M}_\Gamma$：$\mathcal{M}_0 \subseteq \mathcal{M}_1 \subseteq \cdots \subseteq \mathcal{M}_\gamma$，其中有 $\mathcal{M}_i \in \mathcal{M}_\Gamma$， $\mathcal{M}_j \in \mathcal{M}_\Gamma$， $\mathcal{M}_i \subseteq \mathcal{M}_j$ （$i < j$）。若在低端模型 $\mathcal{M}_i$ 中加入论域映射函数引用了高端模型 $\mathcal{M}_j$ 的值域数据，即 $\mathcal{M}_i = (M_i, C_i, R_i, F_i \cup F_{M_j-M_i})$，其中 $F_{M_j-M_i}$: $M_j{}' \to M_i{}'$， $M_i{}' \subseteq M_i$，$M_j{}' \subseteq M_j$。这种情况下，引用的数据效果会通过模型链进行传递，即 $\mathcal{M}_i$ 的输出数据会传递给 $\mathcal{M}_{i+1}$，最终还会传递到 $\mathcal{M}_j$ 的输入，影响到 $\mathcal{M}_j$ 的输出，之后又会再反向传递到 $\mathcal{M}_i$ 的输入，形成循环。称模型链中通过论域映射函数的这种数据循环的引用为**循环引用。**

由定义 4.4，在低端模型 $\mathcal{M}_\mathrm{i}$ 中加入论域映射函数，引用高端模型 $\mathcal{M}_\mathrm{j}$ 中的论域数据，存在数据循环引用的情况。而数据循环引用有可能会造成系统工作不稳定。特别地，在定义 4.4 中，如果令 $\mathrm{i} = \mathrm{j}$，则 $\mathcal{M}_\mathrm{i}$ 将会在输入中引用自身的输出数据，类似于自动调节系统

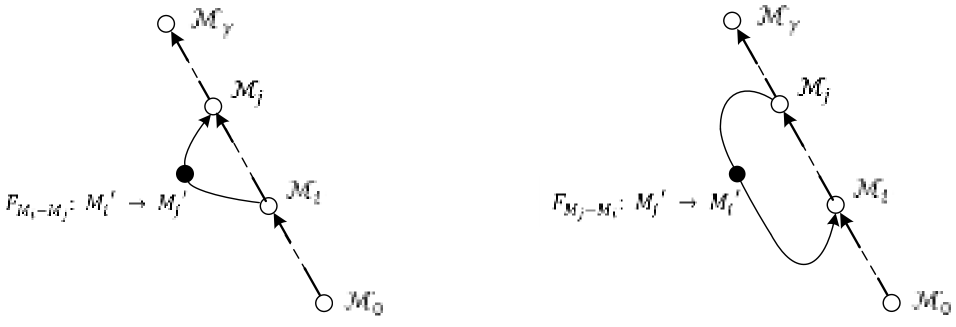
中的“数据反馈”，可能会出现工作不稳定的状态。故有以下定理：

定理4.4： 在模型链中，在低端模型中通过论域映射函数引用高端模型的论域数据，存在数据循环引用情况，有可能造成系统工作不稳定。

由定理 4.4，模型链中在低端模型中通过论域映射函数引用高端模型论域数据的场合，由于存在系统不稳定的因素，故需对系统的技术方案进行验证。对于存在不稳定的系统，应当调整技术方案，确保系统能稳定运行。

c) 模型链中加入论域映射函数后对模型结构的影响

加入论域映射函数后，模型链内部各模型的论域、关系、其它函数未作改变，因此，论域映射函数的加入不影响模型的组合结构。但是数据传递的路径却因此而发生了改变，影响到整个模型组合处理数据的效果。换言之，它影响了数据传递的结构，即影响了数据流结构。图 4.3 表示了在模型链中加入论域映射函数后的数据传递关系。



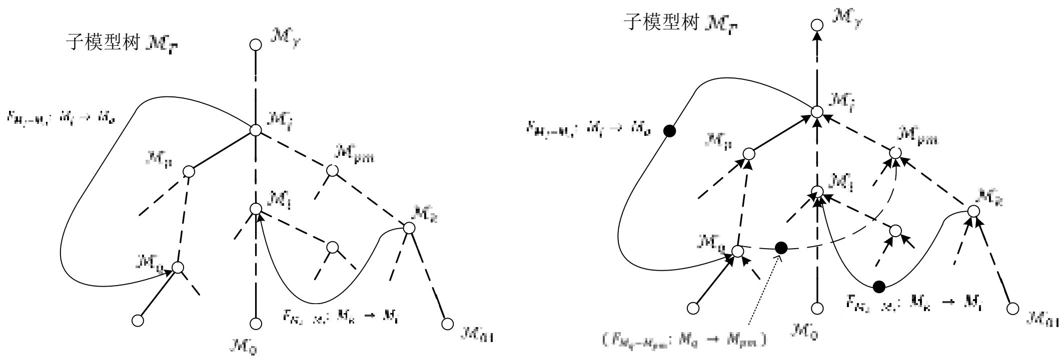
a) 高端模型引用低端模型的数据传递关系　　b) 低端模型引用高端模型数据传递关系

图 4.3 模型链中加入论域映射函数后的数据传递关系

初看起来，图 4.3 与 4.2 并无多大差异，但在不止一处加入论域映射函数时，就不能只考虑原来的模型链了，还应当考虑这之前已加入映射函数后形成的数据传递路径的结构。这时，应当采用数据传递图来考察新加入的论域映射函数对模型链的影响。这个问题在单个模型链中不算突出，但在模型树和模型网络中存在多个模型链的情况下就不能回避了。

2. 论域映射函数对模型树的影响

图 4.4 a）表示了在一个子模型树 $\mathcal{M}_\Gamma$ 中加入论域映射函数的情况。图中，模型 $\mathcal{M}_i$ 加入了论域映射函数 $F_{M_k-M_i}$： M_k → M_i（为方便和表示清晰，这里直接采用了论域表示，而一般情况应为子论域映射），将模型 $\mathcal{M}_k$ 的输出映射到 $\mathcal{M}_i$ 的输入；模型 $\mathcal{M}_q$ 加入了映射函数 $F_{M_j-M_q}$： M_j → M_q，将模型 $\mathcal{M}_j$ 的输出映射到 $\mathcal{M}_q$ 的输入。



a）　模型树中添加论域映射函数　　　b）模型树中添加论域映射函数后的数据传递关系

图 4.4　　模型树中添加论域映射函数示例

由图 4.4 a)还可看出：模型 $\mathcal{M}_k$ 与 $\mathcal{M}_i$ 不在一条模型链上，而模型 $\mathcal{M}_j$ 与 $\mathcal{M}_q$ 处在一条模型链上。根据上面分析，由于模型 $\mathcal{M}_k$ 与 $\mathcal{M}_i$ 不在一条模型链上，不会出现模型数据的循环引用，因而不会影响系统的稳定性；而模型 $\mathcal{M}_j$ 与 $\mathcal{M}_q$ 处在一条模型链上，且 $\mathcal{M}_q$ 为低端模型， $\mathcal{M}_q \subseteq \mathcal{M}_j$, 存在数据循环引用，有可能造成系统不稳定。故有以下推论：

推论4.1： 在模型树中，如果通过论域映射函数进行数据引用的两个模型处在同一个子模型链中，且是将高端模型的输出引用到低端模型输入，则存在数据循环引用，有可能造成系统不稳定。

对可能存在稳定性问题的数据引用，必须要进行验证。如果验证后确认系统存在稳定性问题，则必须要对技术方案进行调整，以确保系统能稳定运行。

如前节所述，在不止一处加入论域映射函数时，新增的映射函数必须考虑已加入映射函数的影响，要从数据传递关系角度去核实是否影响系统的稳定性。如图 4.4 b)所示，在图 4.4 a）模型树已加入映射函数的基础上，又加入了一个论域映射函数 $F_{M_q-M_{pm}}$： $M_q \rightarrow M_{pm}$。如果在加入 $F_{M_q-M_{pm}}$之前，已验证系统是稳定的，但在加入 $F_{M_q-M_{pm}}$ 后，数据又会沿着新的路径 $\mathcal{M}_q \rightarrow \mathcal{M}_{pm} \rightarrow \cdots \rightarrow \mathcal{M}_j$ 进行传递，会出现新的数据循环引用问题，故必须再次进行稳定性验证。而这种新出现的循环引用问题通过数据传递图是易于发现的。故此有以下推论：

推论4.2： 在已加入论域映射函数的模型树中，如果新增的映射函数又与原来的数据传递路径组合形成新的数据通路，出现低端模型引用高端模型论域数据的新路径，则新增论域映射函数后存在新的数据循环引用，有可能造成系统不稳定，需进行稳定性验证。

推论 4.2 指出了在有不止一条模型链的子模型树中会出现新的稳定性问题，需进行验证。如果出现新的不稳定情况，亦需调整技术方案，确保系统稳定运行。

3. 论域映射函数对模型网络的影响

图 4.5 为模型网络 $\mathcal{M}_\Gamma$ 中加入论域映射函数的示例。在 $\mathcal{M}_\Gamma$ 中加入了两个论域映射函数 $F_{M_{p01}-M_{rsj}}$： $M_{p01} \rightarrow M_{rsj}$ 和 $F_{M_{pnj}-M_{qml}}$： $M_{pnj} \rightarrow M_{qml}$。其中， $F_{M_{p01}-M_{rsj}}$ 将模型 $\mathcal{M}_{p01}$的输出映射到模型 $\mathcal{M}_{rsj}$ 的输入，$F_{M_{pnj}-M_{qml}}$ 将模型 $\mathcal{M}_{pnj}$ 的输出映射到模型 $\mathcal{M}_{qml}$ 的输入。图中，模型 $\mathcal{M}_{pnj}$ 与模型 $\mathcal{M}_{qml}$ 不在同一个子模型链上，它们之间的数据不存在

循环引用，因此映射 $F_{M_{pnj}-M_{qml}}$： $M_{pnj} \to M_{qml}$ 不会对系统的稳定性造成影响。而模型 $\mathcal{M}_{p01}$ 与模型 $\mathcal{M}_{rsj}$ 处在同一个子模型链中，且 $\mathcal{M}_{p01}$ 为高端模型，因此，映射函数 $F_{M_{p01}-M_{rsj}}$： $M_{p01} \to M_{rsj}$ 存在数据的循环引用，会对系统的稳定性造成影响。

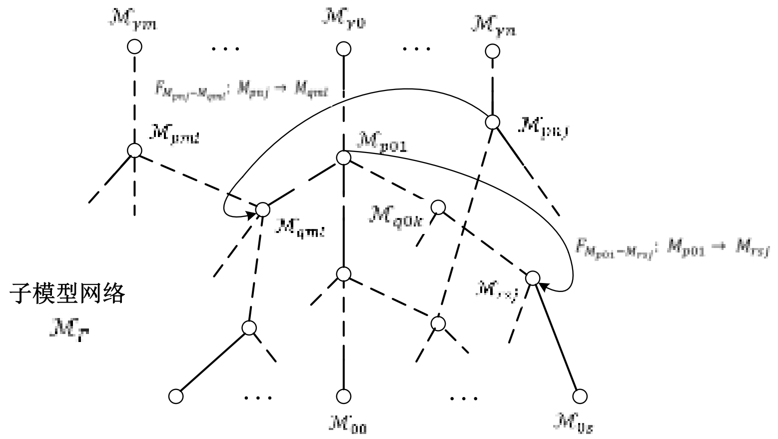


图 4.5　　模型网络中添加论域映射函数示例

根据以上对模型链和模型树中加入论域映射函数的分析，可得到以下推论：

推论4.3：　在模型网络中，如果通过论域映射函数进行数据引用的两个模型处在同一个子模型链中，且是将高端模型的输出引用到低端模型输入，则存在数据循环引用，有可能造成系统不稳定。

在模型网络中，对可能存在稳定性问题的数据引用，必须要进行验证。如果验证后确认系统存在稳定性问题，则必须要对技术方案进行调整，以确保系统能稳定运行。

如前所述，在模型网络中亦存在新增论域映射函数引起新增循环数据路径的问题，图 4.6 表示出了这一情况。

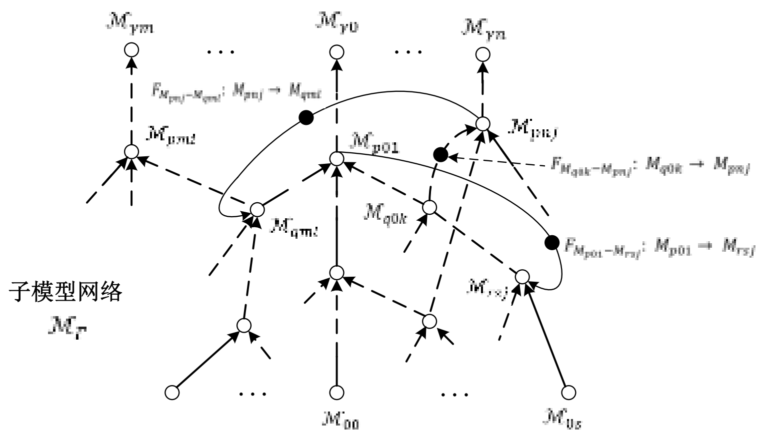


图 4.6　　模型网络中添加论域映射函数后的数据传递图

图 4.6 中，在图 4.5 基础上新增了一个论域映射函数 $F_{M_{q0k}-M_{pnj}}$： $M_{q0k} \to M_{pnj}$，形成了新的数据传递循环路径：$\mathcal{M}_{pnj} \to \mathcal{M}_{qml} \to \mathcal{M}_{p01} \to \mathcal{M}_{rsj} \to \mathcal{M}_{q0k} \to \mathcal{M}_{pnj}$，存在新的不稳定可能，因此需对这个映射函数的加入进行技术验证。有以下推论：

推论4.4：　在已加入论域映射函数的模型网络中，如果新增的映射函数又与原来的数据路径组合形成新的数据传递路径，以致出现低端模型引用高端模型论域数据的新路径，

则新增论域映射函数存在新的数据循环引用，有可能造成系统不稳定，需进行稳定性验证。

综上所述，可总结出以下几点：在模型组合中，如果在同一个子模型链上的低端模型中引用高端模型数据，则存在数据循环引用问题；在模型组合中加入论域映射函数时，采用数据传递图来分析是否存在数据循环引用问题更为方便；对存在数据循环引用的场合，需进行稳定性验证，如果存在系统不稳定因素，则需进行技术方案调整，确保系统能稳定运行。

五. 论域映射函数方法应用展望

1. 对模型组合的应用

前面已经对模型链、模型树、模型网络等模型组合中加入论域映射函数的方法进行了分析。结果表明，1）在一个模型组合中的任意两个模型之间，只要两个模型不是处于同一个子模型链上，均可加入论域映射函数进行数据引用，且不存在稳定性问题；2）即使两个模型处于同一个子模型链上，只要论域映射不是从高端模型把数据映射到低端模型，也不会因论域映射函数的引用而影响系统的稳定性；3）只有当两个模型处于同一个子模型链中，且数据是从高端模型引用到低端模型，才有可能出现数据循环引用问题，进而导致可能出现系统稳定性问题而需要进行验证。因此，在一个结构复杂的模型网络中，为了判断在加入论域映射函数时会不会出现数据的循环引用，采用数据传递图的方法是必要的，如图 4.4 b)与图 4.6 所示。

模型受限并运算实现了模型的组合，定理 3.1 揭示了模型受限并运算方法的本质是在并运算中维持函数和关系的定义域不变，以保持并运算中的函数和关系成立，实现了模型的组合。而论域映射函数方法的引入，极大地方便了模型组合中模型之间处理数据成果的相互利用，使模型组合的性能得到更大的提升。我们期望，论域映射函数方法将会在模型组合中得到广泛应用。

2. 对智能体研究的应用

当前，以大模型为基础的多模态智能系统[14][15][16]和多智能体系统[17][18][19]是人工智能领域的重要发展方向，已取得瞩目的成绩。这两类系统都涉及多个模型间的数据转换、数据交互和相互协同等问题。资料[9]提出的子模型组合方法，从逻辑结构上体现了不同模型间的逻辑关系，对多模态具身智能系统和多智能体系统的研究能起到一定的技术支持作用。而本文提出的论域映射函数方法又对模型之间的数据相互引用给出了可行的意见，因此，将子模型组合方法与论域映射函数方法结合运用，将会对智能体的研究起到较好的支持作用。

对于具身智能体和多智能体系统而言，我们可以将其看成为由若干功能相对独立的子系统或模块构成。从逻辑结构角度看，这些子系统或模块可以看成为一个个的模型，因此，可以把它们组合成更大的系统或模型。本文提出的论域映射函数方法，指出了可以在这些

组成的模型之间进行处理数据的相互引用和应该遵循的逻辑原则，以及对数据循环引用的监测和处理方法。显然，论域映射函数方法对模型之间的数据利用，给出了一种灵活高效的实施方案，这对提高整个系统的数据处理效能，有着重要的作用。本文只是提出了这种跨模型数据引用的原理方案，有待在实际系统进一步运用和完善。

3. 对人工智能基础理论的应用展望

当前，多模态大语言模型，知识图谱，人工智能与多模态数学推理，交叉领域应用，具身智能系统等人工智能领域的技术研究取得了飞速进步和瞩目的成绩[14]--[19]。与此同时，若干基础理论研究工作亦需同步进行。资料[9]提出的模型受限并运算方法，使得可以从数理逻辑角度研究模型的组合问题；而本文提出的论域映射函数方法，使这种模型组合方法的数据处理效能更高，运用效果更好，我们期望它在人工智能的模型构建上得到较广泛的应用。

从数据传递角度看，前述的论域映射函数方法是模型组合中某个模型主动去引用另一个模型的处理数据。在实际应用中，我们可以设想，经一个模型处理后的数据也可以主动向其它有关模型提供支持。例如，一个运动智能体的坐标探测模块获取的自身位置坐标数据，就应该实时主动提交给有关模块使用。在这种情况下，从论域映射角度看，数据传递的逻辑效果与本文前述的主动引用其它模型的数据是等效的。因此，如果在系统中采用向其它模型主动提供处理数据的方案时，同样存在数据循环引用问题，存在影响系统稳定性的因素，只是在这种情况下应该从引用数据的模型处对是否存在数据循环引用进行分析，而主动提交数据的模型不必考虑它的作用和效果。

本文提出的论域映射函数方法是一种论域数据传递的控制方法，可视其为数据引用方法。从实际使用角度看，在大的模型组合中，是否存在功能引用的可能？这值得深入研究。从模型角度看，可作如下描述：设一阶语言 $\boldsymbol{\mathcal{L}}=\{C, R, F\}$ 的两个模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$。现假设由 $\mathcal{M}_1$ 提供论域数据 $M_{1k} \subseteq M_1$，交由 $\mathcal{M}_2$ 处理，借用 $\mathcal{M}_2$ 的函数集 F_2 和关系集 R_2 进行数据变换和推理，最后将结果数据 $M_{2p} \subseteq M_2$ 返回给 $\mathcal{M}_1$。用式子描述为两步实现：首先在 $\mathcal{M}_1$ 中增加映射矢量 $< F_{M_{1k}-M_2}, G_{1-2} >$，其中$F_{M_{1k}-M_2}$为模型 $\mathcal{M}_1$ 提交给 $\mathcal{M}_2$ 的待处理数据，G_{1-2} 为 $\mathcal{M}_1$ 提交给 $\mathcal{M}_2$ 的功能处理要求，包括数据变换和推理方面的要求，即指定 $\mathcal{M}_2$ 中的处理函数和推理关系。$\mathcal{M}_2$ 对 $\mathcal{M}_1$ 提交来的处理要求完成后，再由映射函数 $F_{M_{2p}-M_1}$ 将结果返回给 $\mathcal{M}_1$。这相当于实现了外部模型对 $\mathcal{M}_2$ 内部功能的调用。在复杂的模型组合中这种工作方式可以使模型之间能够相互利用处理资源，是一种值得探索的模型间数据处理的方法。在多智能体的系统中，这种功能引用值得进一步探索研究。

数理逻辑为人工智能系统的研究提供了基础理论的支持。在人工智能的基础技术研究方面，包括建模理论、知识表达与推理、机器学习理论、跨领域应用等，都离不开数理逻辑有关理论的支撑。对于人工智能系统的研究而言，模型受限并运算和论域映射函数方法从数理逻辑模型论角度给出了一种可行的模型组合方法，我们期待它作为人工智能系统模

型构建的一种基础技术而得到广泛应用。

六. 结语

本文在资料[9]模型受限并运算的基础上，通过对模型论域的深入分析，揭示出了模型受限并运算的本质是在模型并运算中维持了函数和关系的定义域不变化，以保证模型并运算的成立。通过进一步的分析，本文提出了论域映射函数方法。该方法是模型组合中模型之间数据引用的一种可行的解决方案，提高了模型数据处理的效能。从整个模型组合看，它可以更充分地利用系统资源，提高整个模型组合的性能，是一个值得在模型构建中采用的系统优化方法。

在人工智能研究领域，基础研究是不可或缺的研究内容。模型受限并运算与论域映射函数方法作为数理逻辑的研究课题，可看作是数理逻辑模型论的一个新的研究方向，它对人工智能模型的构建、多模态具身智能应用、跨领域应用等将起到重要的支持作用。

参考文献

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit U, Jones L,Gomez A N, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, USA: Curran Associates Inc., 2017. 6000–6010.

[2] Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:Human Language Technologies. Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2018. 4171–4186.

[3] Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Florencia B, et al. GPT-4 technical report. arXiv preprint arXiv: 2303. 08774, 2023.

[4] 童云海，陈建龙，DeepSeek 热潮下的双重变革:大模型的技术革新与高校图书馆服务范式的重构，大学图书馆学报，2025 年第 1 期

[5] 孟小峰等，科学发现中的机器学习方法研究. 计算机学报，2023，第 5 期，877-895

[6] Hutson M. Artificial intelligence faces reproducibility crisis. Science, 2018, 359 (6377): 725-726

[7] Khan S, Naseer M, Hayat M, Zamir S W, Khan, F S, Shah M.Transformers in vision: A survey. arXiv preprint arXiv: 2101.01169, 2021.

[8] 赵希顺，简明数理逻辑，科学出版社，2021.11.

[9] 刘志模，刘 徽，王晓山，陈 振，一种树形模型链构建方法及其应用展望，ChinaXiv:202503.00280

[10] 罗里波，模型论及其在计算机科学中的应用，北京师范大学出版社，2012.1.

[11] Michael Huth, Mark Ryan，Lojic in Computer Science ：Modelling and Reasoning about Systems，Second Edition，ISBN 987-7-111-21397-0.

[12] 孙希文，数理逻辑，高等教育出版社，2019.11.

[13] 雷英杰等，模糊逻辑与智能系统，西安电子科技大学出版社，2016.5

[14] Wang J Q, Wu Z H, Li Y W, Jiang H Q, Shu P, Shi E Z, et al.Large language models for robotic

s: Opportunities, challenges,and perspectives. arXiv preprint arXiv: 2401.04334, 2024.
[15] Yang Z Y, Li L J, Lin K, Wang J F, Lin C C, Liu Z C, et al. The dawn of LMMs: Preliminary explorations with GPT4V(ision). arXiv preprint arXiv: 2309.17421, 2023.
[16] 王文晟等，基于大模型的具身智能系统综述，自动化学报，2025 年第 1 期
[17] ZHAO W S, QUERALTA J P, WESTERLUND T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey[C] //Proceedings of 2020 IEEE Symposium Series on Computational Intelligence.[S. l.]:IEEE, 2020:737-744.
[18] LOWE R, WU Y I, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances in Neural Information Processing Systems, 2017, 30: 6382-6393.
[19] IQBAL S, SHA F. Actor-attention-critic for multi-agent reinforcement learning[C] //Proceedings of the 36th International Conference on Machine Learning. [S.l.:s.n.], 2019:5261-5274.

(通讯作者：刘志模　E-mail: hylj2010@sohu.com)

作者贡献声明：

刘志模：提出研究思路，论文起草，终版审定。
陈　振：审改论文，与实体模型进行了对比，对应用前景进行了分析并提出审改意见。
王晓山：审改论文，对论文的应用前景等提出审改意见。
刘　徽：审改论文，对论文的理论论述进行了审改，对应用前景提出审改意见。