第37卷 第14期 计 算 机 工 程 2011年7月
Vol.37 No.14 Computer Engineering July 2011

·人工智能及识别技术· 文章编号：1000—3428(2011)14—0189—03 文献标识码：A 中图分类号：TP18

基于范畴论的粒合成形式化方法

李 鸿[1,2a,2b]

(1. 中国矿业大学信息与电气工程学院，江苏 徐州 221116；2. 宿州学院 a. 智能信息处理实验室；b. 信息工程学院，安徽 宿州 234000)

摘 要：针对粒计算中细粒层到粗粒层的转换问题，提出一种基于范畴论的粒合成形式化方法。以粒结构为对象，将粒结构映射作为态射构成粒结构范畴，利用范畴论的外推实现粒结构合并，并给出相应的粒合成算法GrSA，结合具体案例说明该方法解决粒合成问题的过程。应用结果表明，该方法能够实现粒层之间的转换。

关键词：范畴论；粒结构；粒结构范畴；粒结构映射；粒结构合并；粒合成

Formal Method of Granular Synthesis Based on Category Theory

LI Hong[1,2a,2b]

(1. School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China;
2a. Laboratory of Intelligent Information Processing; 2b. School of Information Engineering, Suzhou University, Suzhou 234000, China)

【Abstract】In order to solve the problem of conversion from the finer level of granularity to the coarser level of granularity, this paper proposes a formal method for granular synthesis based on category theory. It takes the granular structure as the granular object, the granular structure mapping as the granular morphism, then the granular object and the granular morphism composes the granular structure category. It uses the granular pushout to achieve the granular merging, and gets the granular synthesis algorithm GrSA. It illustrates the method with the concrete examples, which solves the granular synthesis issue. This method can realize the conversions between the levels of granularity.

【Key words】category theory; granular structure; granular structure category; granular structure mapping; granular structure merging; granular synthesis

DOI: 10.3969/j.issn.1000-3428.2011.14.063

1 概述

粒计算[1](Granular Computing, GrC)是一种看待客观世界的世界观和方法论，它由美国学者Lin T Y教授于1997年提出[1]。但如何简单快捷地实现粒的分解与合并操作，一直没有很好的解决办法。笔者认为，粒的分解与合并操作应既能从定性的角度分析，又能从定量的角度计算，这样将有利于实现不同粒层之间的来回自由转换，有助于提高粒计算的应用深度和广度。文献[2]建立了9种粒计算模型，其中第8种是基于范畴论的粒计算模型。Lin T Y教授认为，粒计算模型的研究已达到了稳定状态，而且其他8种粒计算模型可由第8种粒计算模型推导得到，因此，他将基于范畴论的粒计算模型作为最后的模型。

笔者在已有的有关粒计算研究成果的基础上，提出了粒集理论[3-6]，对粒集进行了描述，为粒计算提供了一个新的模型，为粒的分解与合并操作奠定了初步基础。本文在文献[2]的基础上，同时借鉴文献[7]的研究思路和方法，提出一种解决粒合成问题的形式化方法。

2 基本概念

2.1 范畴论

范畴论产生于20世纪40年代，是在传统集合论的基础上发展起来的数学理论，其研究重点在于对象之间的关系而非对象的内部结构。因此，它比集合论更适合建立较高抽象层次的模型。范畴论作为一门反映数学各分支共性的理论，为计算机科学提供了一种强有力的工具、思维方法和研究手段，在计算机学科的应用领域不断扩展。

定义1(范畴) 范畴[8](category)是具有所有可能的态射的对象的集合，其具体表示可参见文献[8]。

范畴是从数学的各个领域中概括出来的一种高度抽象的数学系统，而且是基于图的，可以把一个范畴看成一个有向图，图中的节点表示对象，箭头表示态射，每个箭头有一个源节点和目标节点，分别表示论域和余论域(陪域)。

定义2(外推) 如果范畴C中的任意一个形如(f: $C\to A$, g: $C\to B$)的图表都存在余极限(f': $B\to P$, g': $A\to P$)，则称f'为f沿着g的一个外推[8](pushout)，g'为g沿着f的一个外推。

2.2 粒

需要说明的是，本文所讨论的粒，并不是像现代物理学中的中子、质子等实际存在的具体的物质成分，而是一种模型理念。

定义3(粒) 粒[6]是客观存在的各种各样的成分的统称。它是粒计算模型的最基本元素，是一个数据单元或一个计算单元，也可以说粒是粒计算模型的原语。

一个粒可以被解释为许多小颗粒构成的一个大个体，或者理解为由或难以区分、或很相似、或接近、或因某种功能而结合在一起的一个团。粒是一个满足某描述条件的个体对象集合，每一个粒是和它的描述分不开的。一个粒既可以是某个整体的相对独立的一个部分，又可以被看作是由其他一些粒组成的整体。因此，用具有不同粒度的粒描述概念层次

基金项目：安徽高校省级自然科学研究基金资助重点项目(kj2007a129zc)

作者简介：李 鸿(1965－)，男，教授、博士研究生、CCF会员，主研方向：智能计算，算法设计与分析，人工智能

收稿日期：2010-12-23 **E-mail：**lih65@163.com

是非常合理的。可以认为，各种各样的粒及其之间的关系所组成的数学结构抽象地表示了客观世界。

3 粒结构及其操作

要实现粒，首先要了解粒的结构，然后需要建立粒之间的联系，即对粒进行映射，最终实现粒合并。由此可知，粒结构是实现粒集成的基础，粒映射和粒合并是实现粒集成的 2 个主要操作。

但由于粒的构造一直没有一个统一的规范和标准，因此，需要给出一种粒结构的形式化表示形式，在此基础上定义粒映射和粒合并的概念，以便后文利用范畴论对它们进行形式化描述。

定义 4(粒结构) 粒结构(Granular Structure)可定义为如下的四元组形式：$GrS=(G, R, H_G, A_G)$，其中，GrS 表示粒结构；G 表示粒的集合；g_r 表示粒($g_r \in G$)；R 表示粒关系集合；r 表示粒关系($r \in R$)，粒关系描述了粒与粒之间或者属性与属性之间的关系。粒关系可以分为 2 类：分类关系和关联关系。H_G 表示分类关系，是相邻的粒与粒之间的父类、子类等上下位的层次关系，$H_G \subseteq G\times G$；A_G 表示关联关系，是除了上下位层次关系以外非相邻的祖父、兄弟等其他关系，$A_G: R\to G\times G$，它为每个关系确定了作用域。

例 1 设一个粒结构为 $GrS=(\{x_0, x_1, x_2, x_3\}, \{r_1, r_2\}, \{(x_0,x_1), (x_0, x_2), (x_2, x_3)\}, \{(r_1, x_0, x_3), (r_2, x_1, x_2)\})$，则$\{(x_0, x_1), (x_0, x_2), (x_2, x_3)\}$表示：$x_0$ 为 x_1 和 x_2 的父亲，x_2 为 x_3 的父亲；$\{(r_1, x_0, x_3), (r_2, x_1, x_2)\}$表示：$x_0$ 为 x_3 的祖父，x_1 与 x_2 为同一层的兄弟。如图 1 所示。

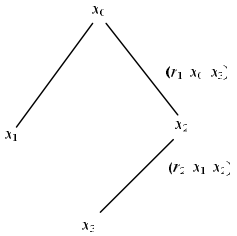


图 1 粒结构

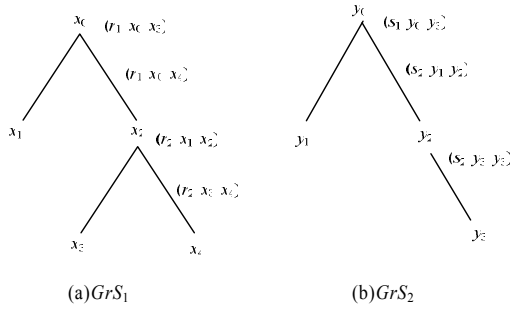
定义 5(粒结构映射) 粒结构映射是指在 2 个粒结构之间按某种法则发现粒实例之间和粒关系之间的相对应的过程。形式化如下：设 2 个粒结构为 $GrS=(G, R, H_G, A_G)$和 $GrS'=(G', R', H_{G'}, A_{G'})$，粒结构映射定义为粒实例映射和粒关系映射构成的形式化的二元组形式：

(f, g): $GrS\to GrS'$或(f, g): $G\times R\to G'\times R'$

其中，f: $G\to G'$称为粒实例映射；g: $R\to R'$称为粒关系映射，且对所有的 $G_1,G_2 \in G$ 满足条件：(1)如果$(G_1, G_2) \in H_G$，则$(f(G_1), f(G_2)) \in H_{G'}$；(2)如果$(G_1,G_2) \in A_G(R)$，则$(g(G_1), g(G_2)) \in A_{G'}(g(R))$。条件(1)保持了粒的层次结构，条件(2)保持了粒实例之间的关联关系。下面通过例子来说明粒结构映射的具体情况。后文将粒结构映射简称为粒映射。

例 2 设 2 个粒结构为 $GrS_1=(\{x_0, x_1, x_2, x_3, x_4\}, \{r_1, r_2\}, \{(x_0, x_1), (x_0, x_2), (x_2, x_3), (x_2, x_4)\}, \{(r_1, x_0, x_3), (r_1, x_0, x_4), (r_2, x_1, x_2), (r_2, x_3, x_4)\})$, $GrS_2=(\{y_0, y_1, y_2, y_3\}, \{s_1, s_2\}, \{(y_0, y_1), (y_0, y_2), (y_2, y_3)\}, \{(s_1, y_0, y_3), (s_2, y_1, y_2) , (s_2, y_3, y_3)\})$，其具体结构可参考例 1 的说明。

若粒实例映射为：$x_0\mapsto y_0, x_1\mapsto y_1, x_2\mapsto y_2, x_3\mapsto y_3, x_4\mapsto y_3$，粒关系映射为：$r_1\mapsto s_1, r_2\mapsto s_2$，则两者共同构成的二元组形式$(f, g)$: $\{x_0, x_1, x_2, x_3, x_4\}\times\{r_1, r_2\}\to\{y_0, y_1,y_2,y_3\}\times\{s_1, s_2\}$即为粒映射。这个粒映射既保持了粒的层次关系，也保持了粒之间除父子关系之外的其他关系。例如，在粒 GrS_1 中，x_0 是 x_1 和 x_2 的父粒。相应地，在 GrS_2 中，$f(x_0)=y_0$ 是 $f(x_1)=y_1$ 和 $f(x_2)=y_2$ 的父粒；粒 GrS_1 中的(r_1,x_0, x_3)与 GrS_2 中的$(g(r_1), f(x_0), f(x_3))=(s_1, y_0, y_3)$形成映射。如图 2 所示。



(a)GrS_1 (b)GrS_2

图 2 粒结构映射

定义 6(粒结构合并) 粒结构合并建立在粒结构映射基础上，是将 $n(n\geqslant 2)$个相关的粒合并成一个新的粒的过程，新粒是这 n 个粒的并集，不仅包括原来 n 个粒的相似部分，也包含了不同的部分。

4 基于范畴论的粒合成算法

考虑到粒结构就是粒实例以及粒实例之间的关系所构成的有向图，可以把这种有向图看作一个粒结构范畴。粒的合成 2 个关键操作——粒结构映射和粒结构合并可以由范畴论的相关性质来描述。因此，即可将粒结构作为对象，粒结构映射作为态射，两者共同构成了一个粒结构范畴，这样就可以通过范畴论的外推来实现粒合并。

4.1 粒结构范畴

定义 7(粒结构范畴) 以粒结构作为对象类，以粒结构映射为态射形成的范畴，则称为粒结构范畴，记作 $GrSC$，而其中对象类记作 $Ob(GrSC)$，粒态射类记作 $Mor(GrSC)$。

在粒结构范畴中，单位态射是粒结构类上的空映射；如果粒态射$(f_1, g_1), (f_2, g_2)$复合有意义，则$(f_1, g_1)\circ(f_2, g_2)$也是一个粒态射，$(f_1, g_1)\circ(f_2, g_2)\in Mor(GrSC)$且是唯一的；如果$(f, g)$: $A\to B$ 是粒结构 A 到 B 的映射，(f, g)与空映射复合仍然是(f, g)，即$(f, g)\circ ID_B=ID_A\circ(f, g)=(f, g)$。

由定义 7 可知，在粒结构范畴 $GrSC$ 中，其重点不是粒对象类中的粒结构的具体形式，而是粒对象类之间的关系，也就是粒之间的映射。

4.2 粒外推

由定义 2 可以看出，范畴论中外推的作用是形成融合和合并，因此，通过范畴论的外推性质实现粒合并是很自然、形象的。下面根据范畴论中的外推定义粒的外推，以此来实现粒的合并。

定义 8(粒外推) 设粒结构范畴中有 3 个粒结构：GrS_1，GrS_2 和 GrS，粒态射(f_1, g_1): $GrS\to GrS_1$ 和(f_2, g_2): $GrS\to GrS_2$，如果存在粒结构 GrS' 及态射$(f_1,g_1)'$: $GrS_1\to GrS'$和$(f_2, g_2)'$: $GrS_2\to GrS'$，满足条件：

(1)$(f_1, g_1)'\circ(f_1, g_1)=(f_2, g_2)'\circ(f_2, g_2)$

(2)对任一其他粒 GrS''的态射$(f_1, g_1)''$:$GrS_1\to GrS''$和$(f_2, g_2)''$: $GrS_2\to GrS''$，存在唯一的射(f, g): $GrS'\to GrS''$，使$(f, g)\circ(f_1, g_1)'=(f_1, g_1)''$，$(f, g)\circ(f_2, g_2)'=(f_2, g_2)''$，则称 GrS'及态射$(f_1,g_1)'$和$(f_2, g_2)'$是 GrS_1 和 GrS_2 的外推。如图 3 所示。

可以直观地看出，GrS'就是由 GrS_1 和 GrS_2 合并而来的。因此，粒外推其实质就是粒的合并，也就是合并粒。基于此，本文提出一种粒合成算法 GrSA。

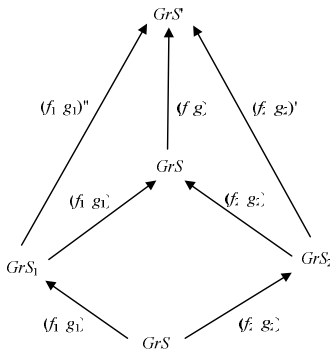


图 3　粒外推

4.3　粒合成算法 GrSA

从图 3 还可以看出，为了将 GrS_1 和 GrS_2 合并，需要一个辅助的粒结构 GrS。为了使粒合成算法具有一般性，这个辅助的粒结构 GrS 应该是最简单的粒层次结构。

定义 9(最简单的粒层次结构) 将只有 2 个粒实例和一个父子关系组成的粒结构称之为最简单的粒层次结构，具体表示形式为 $GrS=(\{w_0, w_1\}, \{u_1\}, \{(w_0, w_1)\}, \{(u_1,w_0, w_1)\})$。如图 4 所示。

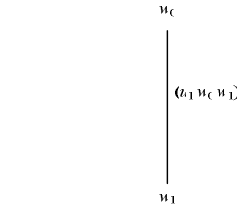


图 4　最简单的粒层次结构

这个最简单的粒层次结构可以看作任意 2 个较复杂的粒结构的交集，即 2 个较复杂的粒结构的相同部分。下面给出基于粒外推的粒合成算法。

4.3.1　粒合成算法的基本思想

从最简单的粒层次结构出发，首先将 2 个较复杂的粒结构的相同部分添加到新粒中，然后不断添加 2 个需要合并的粒的其余部分，最后将 2 个较复杂的粒结构的层次关系添加到新粒的层次关系，这样就可将 2 个不同的粒合并。

4.3.2　粒合成算法的基本步骤

设需要合并的 2 个粒为 GrS_1 和 GrS_2，$GrS_i=(G_i, R_i, H_{Gi}, A_{Gi})$，其中，$i$=1,2。合并结果粒为 $GrS'=(G', R', H_{G'}, A_{G'})$，辅助粒为 $GrS=(\{w_0, w_1\}, \{u_1\}, \{(w_0, w_1)\}, \{(u_1,w_0, w_1)\})$是粒 GrS_1 和 GrS_2 的最小层次结构交集。初始条件 $GrS'=GrS_\varepsilon=(\varnothing, \varnothing, \varnothing,\perp)$是一个空粒，其中，$\varnothing$ 表示集合为空；$\perp$表示函数(这里是 f_i', g_i')在所有的域内无定义。粒合成算法基本步骤如下：

步骤 1 将 GrS_1 和 GrS_2 的共同部分 GrS 添加到 GrS'中。

对最简单的层次结构 GrS 中粒集 G 的每一个实例 g_r，有 $f_1:g_r\mapsto g_{r1}, g_1:u_1\mapsto r_1$ 和 $f_2:g_r\mapsto g_{r2}, g_2:u_1\mapsto s_1$，定义映射 $f_1':g_{r1}\mapsto g_r', g_1': r_1\mapsto t_1$ 和 $f_2':g_{r2}\mapsto g_r', g_2':s_1\mapsto t_1$，则有 $f_1':f_1(g_r)\mapsto g_r', g_1': g_1(u_1)\mapsto t_1$ 和 $f_2': f_2(g_r)\mapsto g_r', g_2':g_2(u_1)\mapsto t_1$ 成立。也就是对 f_i'与 f_i, g_i'与 g_i 进行了复合运算 $f_i'\circ f_i$，$g_i'\circ g_i$，其中，i=1,2，并且 $f_1'\circ f_1$ 和 $f_2'\circ f_2$，$g_1'\circ g_1$ 和 $g_2'\circ g_2$ 分别相等。从而在 G'中相应增添一个新的粒实例 g_r'，g_r'由 $f_i(g_r)$定义。因此，$f_i'(g_r)$和新增的粒实例 g_r'之间建立了映射。这就保证了 GrS'中的每一个粒实例都是 GrS_1 和 GrS_2 的交集，也就是将 GrS_1 和 GrS_2 中相同的 2 个粒实例作为 GrS'中的一个新粒添加到 GrS'中。做下面的并运算：$G':=G'\cup g_r'$，$f_1':=f_1'\cup(f_1(g_r)\mapsto g_r')$，$f_2':=f_2'\cup(f_2(g_r)\mapsto g_r')$，$g_1':=g_1'\cup(g_1(u_1)\mapsto t_1)$，$g_2':=g_2'\cup(g_2(u_1)\mapsto t_1)$。

步骤 2 将 GrS_1 中不是 f_1 映像的其余的粒实例也添加到 GrS'中。

由步骤 1，此时的 f_1'是：对任意 GrS_1 中不是 f_1 映像的粒实例 g_r，有 $f_1(g_r)\mapsto g_r$；此时的 g_1'应是：对任意 GrS_1 关系集中没有在 g_1 和步骤 1 中用到的关系 r，有 $g_1(r)\mapsto r$。进行以下并运算：$G':=G'\cup g_r$，$f_1':=f_1'\cup(f_1(g_r)\mapsto g_r)$ ，$g_1':=g_1'\cup(g_1(r)\mapsto r)$。

步骤 3 将 GrS_2 中不是 f_2 映像的其余的粒实例也添加到 GrS'中。方法同步骤 2。

步骤 4 将层次关系 H_{G1} 增添到 $H_{G'}$中。

对任意 H_{G1} 中父子结构(c, d)，经步骤 2 中得到的 f_1'作用后得到$(f_1'(c), f_1'(d))$，做并运算：$H_{G'}:=H_{G'}\cup(f_1'(c), f_1'(d))$。

步骤 5 将层次关系 H_{G2} 增添到 $H_{G'}$中。方法同步骤 4。

步骤 6 将关联关系 A_{G1} 增添到 $A_{G'}$中。

对任意 A_{G1} 中关联关系(r, c, d)，经步骤 2 中得到的 f_1'和 g_1'作用后得到$(g_1'(r), f_1'(c), f_1'(d))$，做并运算：$A_{G'}:=A_{G'}\cup(g_1'(r),f_1'(c), f_1'(d))$。

步骤 7 将关联关系 A_{G2} 增添到 $A_{G'}$中。方法同步骤 6。

由于态射是保持结构的，因此 $f_i'\circ f_i$，其中，i=1,2，保证了 $H_{G'}$是包含 H_{Gi} 的，它也保证了 GrS'是粒的外推且集合最小，任何其他能替代 GrS'的集合均较大。集合 R'和函数 $A_{G'}$ 也是如此。

4.3.3　粒合成算法描述

输入 $(f_1, g_1):GrS\to GrS_1, (f_2, g_2): GrS\to GrS_2$

输出 $(f_1', g_1'): GrS_1\to GrS', (f_2', g_2'): GrS_2\to GrS'$

初始条件：$GrS'=GrS\varepsilon, f_i'=\perp, g_i'=\perp$

(1)对任意一个 $g_r\in G$，存在 g_r'，这里 $g_r'\square(G_1\cup G_2)$,有：$f_1(g_r)\square g_r'$ 且 $f_2(g_r)\square g_r'$, $g_1(u_1)\mapsto t_1$ 且 $g_2(u_1)\square t_1$

$G':=G'\cup g_r', f_1':=f_1'\cup(f_1(g_r)\mapsto g_r'), f_2':=f_2'\cup(f_2(g_r)\mapsto g_r'), g_1':=g_1'\cup(g_1(u_1)\mapsto t_1), g_2':=g_2'\cup(g_2(u_1)\mapsto t_1)$

(2)对任意一个 $g_r\in G_1$，这里 g_r 不是 f_1 的映像，有：$f_1(g_r)\mapsto g_r$ 且 $g_1(r)\mapsto r$

$G':=G'\cup g_r, f_1':=f_1'\cup(f_1(g_r)\mapsto g_r), g_1':=g_1'\cup(g_1(r)\mapsto r)$

(3)对任意一个 $g_r\in G_2$ ，这里 g_r 不是 f_2 的映像，有：$f_2(g_r)\mapsto g_r$ 且 $g_2(r)\mapsto r$

$G':=G'\cup g_r, f_2':=f_2'\cup(f_2(g_r)\mapsto g_r), g_2':=g_2'\cup(g_2(r)\mapsto r)$

(4)对任意一个 $(c, d)\in H_{G1}, H_{G'}:=H_{G'}\cup(f_1'(c), f_1'(d))$

(5)对任意一个 $(c, d)\in H_{G2}, H_{G'}:=H_{G'}\cup(f_2'(c), f_2'(d))$

(6)对任意一个$(r, c, d)\in A_{G1}$, $A_{G'}:=A_{G'}\cup(g_1'(r), f_1'(c), f_1'(d))$

(7)对任意一个$(r, c, d)\in A_{G2}$, $A_{G'}:=A_{G'}\cup(g_2'(r), f_2'(c), f_2'(d))$

4.4　粒合成算法应用

例 3 以例 2 中的粒 GrS_1 和 GrS_2 为例，说明粒合成算法 GrSA 的应用。

设辅助粒结构 $GrS=(\{w_0, w_1\}, \{u_1\}, \{(w_0, w_1)\}, \{(u_1, w_0, w_1)\})$，态射$(f_1, g_1):GrS\to GrS_1$ 和$(f_2, g_2):GrS\to GrS_2$ 分别定义为：$f_1:w_0\mapsto x_0, w_1\mapsto x_3, g_1: u_1\mapsto r_1, f_2: w_0\mapsto y_0, w_1\mapsto y_3, g_2: u_1\mapsto s_1$。

GrSA 算法在本例中的应用过程既可用文字详细描述，也可以用图示说明，限于篇幅，本文只将结果用图示表示，见图 5。

例 3 说明了算法 GrSA 如何从已知的 2 个粒创建一个新

(下转第 194 页)