arXiv:2305.03432v1 [cs.LO] 5 May 2023

寻找通往罗马的正确道路：效应导向图转换扩展版

延斯·科西奥尔[1],丹尼尔·施特吕贝尔[2,3], 加布里埃莱·坦策尔[1], 和施特芬·兹沙勒[4]

[1] 马尔堡菲利普大学，德国马尔堡
{taentzer,kosiolje}@mathematik.uni-marburg.de[2] 瑞典哥德堡的查尔姆斯｜理工大学danstru@chalmers.se[3] 荷兰奈梅亨拉德堡德大学[4] 英国伦敦的伦敦国王学院
szschaler@acm.org

摘要。 许多图变换应用要求规则在改变图的同时不引入新的一致性违规。设计此类规则时，自然会考虑期望的结果状态，即期望的效应，而不是实现这一结果所需的具体步骤；这些步骤可能因具体的规则应用上下文而异。现有的图变换方法要么要求针对每一种可能的应用上下文分别编写规则，要么无法限制规则所产生的最大变化。我们提出效应导向图变换，将规则的语义从指定操作转变为表示期望效应。单条效应导向规则即可编码大量派生的经典规则。执行哪些潜在操作取决于应用上下文；最终，条条道路通罗马。如果潜在操作要删除（创建）的图元素已经不存在（存在），则无需执行该操作，因为期望的结果已经实现。我们对效应导向图变换进行了形式化定义，说明了如何在不显式枚举所有派生经典规则的情况下计算匹配，并报告了在 Henshin 中对效应导向图变换原型实现的研究。

关键词： 图变换 · 双推送出方法 · 保持一致性的变换

1 引言

图变换的应用，例如模型同步 [15,14,21] 或基于搜索的优化 [6,20] ，需要图变换规则将图的变更与修复 [28] 操作结合起来，以确保变换能够维持甚至提升一致性 [22]。对于任意给定的图约束，

通常都有许多不同的方式会违反该约束，因此需要略有不同的具体变更来修复违规。因此，与其关注修复型图变换规则所需的具体变换，我们往往更容易思考其期望达到的效果。我们希望能够使图达到某种特定状态——根据特定图元素的存在或缺失来定义（即效果）——即使在不同情况下，实现这一目标所需的具体变更各不相同。

现有的图变换方法很难在不明确指定所需具体变更集合的情况下，精确捕捉规则的效果。例如，双推送出方法（*DPO*）用于图变换 [11,12]已被广泛接受为图和模型变换规则在实践中的一种基础形式语义，因为它简单直观：一个变换规则只需指定应用时要删除和创建哪些图元素；也就是说，一个规则恰好规定了所有需要执行的操作。对于图修复而言，这实际上迫使人们指定约束可能被违反的每一种方式，以及在该情况下要应用的具体变更，从而可以根据上下文应用正确的规则。

在另一端，双拉回方法 [19] 则更加灵活。在这种方法中，规则只指定最小变更。然而，没有办法从操作层面约束可能发生的最大变更。

目前还没有一种图变换方法，能够在无需指定每一项待执行操作的情况下，简洁且精确地规定规则的效应。本文提出效应导向图转换的概念。在这种方法中，图变换规则编码了数量可能很大的诱导规则。为此，我们区分了两类操作：任何符合该规则的变换都必须执行的基本操作，以及只有在实现预期规则效应所需时才必须执行的潜在操作。根据应用上下文的不同，执行的操作集合也会有所不同——条条道路通罗马。我们提供了一种算法，可根据上下文选择恰当的操作集合，而无需显式枚举所有可能性——我们能高效地找到通往罗马的正确道路。

因此，本文作出了以下贡献：

1. 我们定义了效应导向图变换规则这一新概念，并讨论了针对此类规则的不同一致匹配概念；2. 对于效应导向变换规则，我们提供了一种算法，可根据部分匹配构造完整匹配和变换。该算法十分高效，因为它避免了显式计算并匹配所有诱导规则；3. 我们介绍了在 Henshin 中实现效应导向变换的原型；4. 我们将该方法与现有的图变换方法进行比较，表明它确实提供了新的表达能力。

本文其余部分的结构如下。首先，我们介绍一个贯穿示例（第2节），并简要回顾基础预备知识（第3节）。第4节介绍了面向效果的规则与变换，以及构造匹配的几种概念。

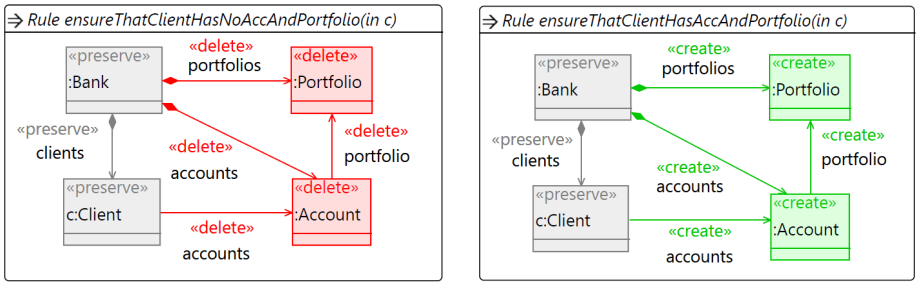

图1. 以 Henshin 的集成可视化语法展示的示例规则 [1,29]。规则的左侧由所有红色和灰色元素组成（并额外标注为 删除 或 保留），右侧则由所有绿色（并额外标注为 创建）和灰色元素组成。

第5 节详细介绍了一种为面向效果的规则构造匹配的算法，并报告了在 Henshin 中的原型实现。第6节讨论现有应用如何从面向效果的规则与变换中受益，并将我们新提出的图变换方法与其他可用于实现这些目标的方法进行比较。最后，我们在第7节总结全文。两个附录分别给出一些补充结果和说明（附录 A），以及对形式化论述的证明（附录B）。

2 贯穿示例

我们采用众所周知的银行示例 [23]，并稍作调整，以说明我们新引入的效应导向变换概念。假设该示例的上下文由一个针对银行领域的元模型来描述，该元模型形式化为一个类型图（未展示），其中银行拥有客户、账户和投资组合。客户可以拥有账户，而账户可以关联到投资组合。

设想这样一种情形：需要确保 客户 拥有一个 账户，且该账户关联着一个 投资组合。为了以基于规则的方式实现这一条件，同时避免创建不必要的元素，至少需要三条规则（以及一个用于协调规则应用的程序）：一条规则用于检查 客户 是否已经拥有一个关联着投资组合 的 账户；一条规则用于为该客户现有的 账户 添加一个投资组合；还有一条规则用于创建所有必需的结构。最后这条规则在图 ensureThatClientHasAccAndPortfolio 中展示为规则，见图 1。

如果要删除账户和投资组合，而它们属于某个客户，也会出现类似的问题。图 ensureThatClientHasNoAccAndPortfolio 中的规则1 会删除整个结构，而要删除未与账户关联的投资组合，还需要额外的规则。一般来说，所需规则的数量及其协调的复杂程度，取决于需要同时出现并因此需要创建（删除）的结构的规模。

借助我们提出的效应导向规则和变换这一新概念，我们能够通过让规则的语义取决于其应用时所处的上下文，使用单条规则来描述所有期望行为。

具体而言，如果在面向效果的语义下，将规则 ensureThatClientHasAccAndPortfolio 应用于某个客户，则可以匹配已有的账户和/或投资组合（而不是创建它们），并且只创建其余尚不存在的部分。因此，我们将账户和投资组合的创建操作称为潜在动作，只有在相应元素尚不存在时才会执行这些操作。同样，在面向效果的语义下应用规则 ensureThatClientHasNoAccAndPortfolio 时，可以什么也不删除（如果匹配到的客户没有账户和投资组合），也可以只删除账户（该账户没有投资组合）。因此，删除操作是潜在动作，只有在相应元素存在时才会执行。我们允许效应导向规则中的部分操作为强制操作。请注意，在 Henshin 规则中，我们需要提供额外注释，以区分强制操作和潜在动作。我们将为这种匹配定义不同的策略，以适用于不同的应用场景。

3 预备知识

本节将简要回顾基本的预备知识。在全文中，我们使用类型化图，并将属性和类型继承的处理留待后续工作。为简洁起见，我们省略嵌套图条件及其沿态射的 [17,13]Shift。我们还省略范畴论中的基本概念；特别是，省略关于粘合范畴的标准事实 [24,12]（类型化图是粘合范畴的一个例子）。尽管这些内容在我们的证明中是必要的，但无需了解它们也能理解本文的核心思想。

定义 1（图；图态射）。一个图 $G = (V_G, E_G, src_G, tar_G)$ 由一个节点集合（或顶点） V_G、一个边集合 E_G以及源函数和目标函数src_G、tar_G: $E_G \to V_G$ 组成；这些函数为每条边指定一个源节点和一个目标节点。

从图 G 到图 H 的一个图态射 $f = (f_V, f_E)$ 是一对函数 f_V: $V_G \to V_H$ 和 f_E: $E_G \to E_H$，二者都与源函数和目标函数交换，即满足 $src_H \circ f_E = f_V \circ src_G$ 以及 $tar_H \circ f_E = f_V \circ tar_G$。如果 f_V 和 f_E 都是单射/满射/双射，则称图态射为相应类型。我们用带钩箭头表示单射态射，即 f: $G \hookrightarrow H$。

类型化有助于赋予图以含义；类型图 提供元素可用的类型，而态射则将类型化图 中的元素映射到这些类型。

定义 2（类型图、类型化图）。给定一个固定的图 TG（即类型图），一个类型化图 $G = (G, type_G)$（基于 TG）由一个图 G和一个态射 $type_G$: $G \to TG$ 组成。两个类型化图 G 和 H（均基于同一个类型图 TG）的一个类型化态射 f: $G \to H$ 是满足 $type_H \circ f = type_G$ 的图态射。

本文中，我们假设所有图均以给定的类型图为类型，且所有态射均为类型态射。不过，为了（记号上的）简洁起见，我们将类型信息视为隐含内容，只称其为图和态射。此外，本文讨论的所有图都是有限的。

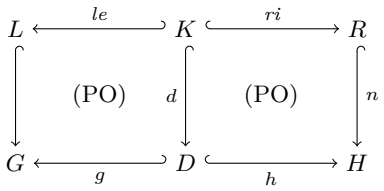

图 2. 双推送出方法中的基于规则的变换

定义 3（规则与变换）。一个规则 $\rho = (p,\ ac)$ 由一个普通规则 p 和一个应用条件ac组成。普通规则是类型化图之间单射态射构成的一个跨度 $p = (L \xhookleftarrow{\text{左侧}} K \xhookrightarrow{\text{右侧}} R)$；其中的图分别称为左侧（*LHS*）、接口和右侧（*RHS*）。应用条件 ac 是一个关于 *[17]* 的嵌套条件 L。

给定规则 $\rho = (L \xhookleftarrow{le} K \xhookrightarrow{ri} R, ac)$ 和单射态射 $m\colon L \hookrightarrow G$，一个（直接）变换 $G \Longrightarrow_{\rho,m} H$ 从 G 到 H （在双推送出方法中）由图 *2* 中的图示给出，其中两个方框都是推出，且 m 满足应用条件 ac，记作 $m \models ac$。如果这样的变换存在，则态射 m 称为匹配，规则 ρ 在匹配 m处可应用；此时， n 称为该变换的余匹配。从规则的左侧到某个图 G的单射态射 $m\colon L \hookrightarrow G$，且满足 $m \models ac$，称为预匹配。

要使规则可在预匹配 m 处可应用， $m \circ le$ 必须存在一个推送出补；对于类图结构的范畴，可以用悬挂条件 [12, 给出一种初等刻画。事实 3.11]：规则可在预匹配 m 处应用，当且仅当 m 不会将 L 中待删除的节点映射到 G 中一个关联着不会同时被删除的边的节点。

应用条件可以沿态射进行“移位”，从而保留其语义 [13, 引理 3.11]。我们在定义子规则 时预设了这一操作，而不再重复说明。我们的子规则概念是核 和多重规则 概念的简化形式 [16]。

定义 *4*（子规则）。给定一条规则 $\rho = (L \xhookleftarrow{le} K \xhookrightarrow{ri} R, ac)$， ρ的子规则是一条规则 $\rho' = (L' \xhookleftarrow{le'} K' \xhookrightarrow{ri'} R', ac')$，以及一个子规则嵌入 $\iota\colon \rho' \hookrightarrow \rho$，其中 $\iota = (\iota_L, \iota_K, \iota_R)$，且 $\iota_X\colon X' \hookrightarrow X$ 是针对 $X \in \{L, K, R\}$ 的单射态射，使得图 *3* 中的两个方块都是拉回，并且 $ac \equiv Shift(\iota_L, ac')$。

4 面向效果的规则与变换

面向效果的语义背后的直觉是：规则规定的是应用该规则后应当呈现的状态，而不是需要执行的操作。在本节中，我们将深入探讨这一方法。我们将引入面向效果的规则 作为一种。

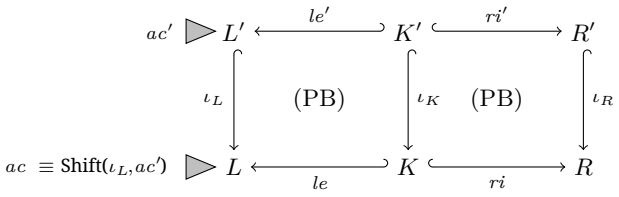

图 3. 规则 ρ的子规则 ρ'

以紧凑的方式表示整组诱导规则。所有诱导规则都共享一个基规则 作为子规则（规定必须执行的操作），但对面向效果的规则所允许的潜在操作 采取不同的选择。第二步中，我们为面向效果的规则制定语义；实际应用哪条诱导规则，取决于面向效果变换的更大上下文。在这里，我们实现了这样一种思想：如果存在合适的元素，就执行面向效果规则中的潜在删除，否则可以跳过。与此相反，只有在尚不存在合适元素时，才执行潜在创建。这样既能使删除数量最大化，又能使创建数量最小化。我们提出了两种方式，以形式化定义这种“极大性”和“最小性”。

4.1 效应导向规则作为规则集的表示

在效应导向规则中，极大规则 通过潜在动作扩展基规则 。在此处以及下文中，我们均假设规则的左右态射以及规则之间的态射（例如子规则嵌入）实际上都是包含映射。这并不会失去一般性（因为总能通过对元素进行重命名来实现所需情形），但能显著简化表述。

定义 5（效应导向规则）。效应导向规则 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$是一个规则 $\rho_{\mathrm{m}} = (L_{\mathrm{m}} \xleftarrow{le}{}^{\mathrm{m}} K_{\mathrm{m}} \xrightarrow{ri}{}^{\mathrm{m}} R_{\mathrm{m}}, ac_{\mathrm{m}})$，称为极大规则；它还包含一个子规则 $\rho_{\mathrm{b}} = (L_{\mathrm{b}} \xleftarrow{le}{}^{\mathrm{b}} K_{\mathrm{b}} \xrightarrow{ri}{}^{\mathrm{b}} R_{\mathrm{b}}, ac_{\mathrm{b}})$，称为基规则，以及一个子规则嵌入$\iota$： $\rho_{\mathrm{b}} \hookrightarrow \rho_{\mathrm{m}}$，使得 $K_{\mathrm{b}} = K_{\mathrm{m}}$（且 ι_K 是恒等映射）。

极大规则 ρ_{m} 的潜在删除是 $(L_{\mathrm{m}}|K_{\mathrm{m}}) \setminus L_{\mathrm{b}} = L_{\mathrm{m}} \setminus L_{\mathrm{b}}$中的元素；同理，其潜在创建是 $(R_{\mathrm{m}} \setminus K_{\mathrm{m}}) \setminus R_{\mathrm{b}} = R_{\mathrm{m}} \setminus R_{\mathrm{b}}$中的元素。此处及下文中，“$\setminus$”表示节点集合和边集合上的逐分量差。

虽然要求 $K_{\mathrm{b}} = K_{\mathrm{m}}$ 会限制面向效果的规则的表达能力，但这已足以满足我们的目的，并且能简化其匹配的定义。如果在匹配过程中，潜在操作会与潜在接口元素争夺可映射到的元素，那么建立匹配极大性的概念就会更加复杂。

示例 *1*。 我们将图 1 中的规则视为效应导向规则的极大规则。在每种情况下，对于应选择哪个子规则

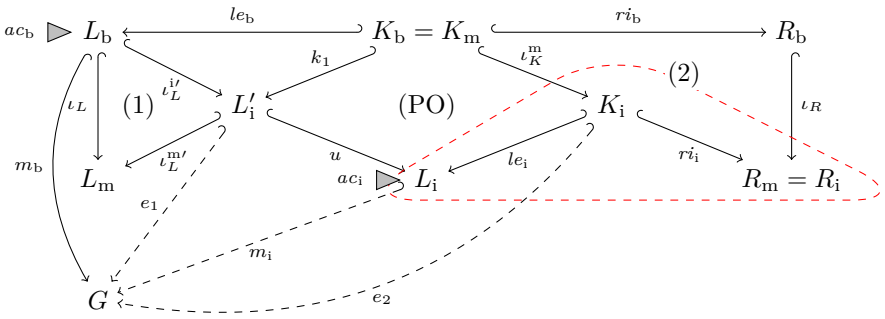

图4. 诱导规则的构造（以红色虚线边框标示）及其匹配

作为规则的基规则，都存在不同的可能性。我们约定，效应导向规则的基规则与极大规则的接口相一致，这意味着在每种情况下，基规则至少包含需要保留的接口。这种最小选择使所有删除和创建都成为可能。

具体而言，对于规则 ensureThatClientHasAccAndPortfolio，可以假设所有待创建元素都只属于极大规则，并表示潜在创建。另一种可能的做法是，将创建账户（连同其入边）的规则视为基规则，而将创建投资组合及其入边视为潜在创建。还可以进行进一步组合。

效应导向规则 ρ_e 表示一组诱导规则。诱导规则通过将极大规则中的潜在删除和潜在创建添加到 ρ_e 的基规则中构造而成。不过，我们要求每条诱导规则的右侧都与极大规则的右侧相同。在诱导规则中，通过将潜在创建也纳入接口，可以省略这些创建。这样可以确保即使并未执行所有潜在创建，应用诱导规则后仍能保持极大规则右侧所表示的状态。

定义6（诱导规则）。给定一个效应导向规则 $\rho_\mathrm{e} = (\rho_\mathrm{b}, \rho_\mathrm{m}, \iota)$，每个规则 $\rho_\mathrm{i} = (L_\mathrm{i} \xleftarrow{le} K_\mathrm{i} \xrightarrow{ri} R_\mathrm{i}, ac_\mathrm{i})$ 如果按以下方式构造（见图*4*的上半部分），则是它的诱导规则：*4):*

1. 存在一个分解 *(1)* $\iota_L = \iota_L^{\mathrm{m}\prime} \circ \iota_L^{\mathrm{i}\prime}$，将 $\iota_L\colon L_\mathrm{b} \rightarrow L_\mathrm{m}$ 分解为两个单射$\iota_L^{\mathrm{m}\prime}$ 和 $\iota_L^{\mathrm{i}\prime}$。*2.* 存在一个分解 *(2)* $\iota_R \circ ri_\mathrm{b} = ri_\mathrm{m} = ri_\mathrm{i} \circ \iota_K^\mathrm{m}$，将 $ri_\mathrm{m}\colon K_\mathrm{m} \rightarrow R_\mathrm{m}$ 分解为两个单射 ri_i 和 ι_K^m，使得方形 *(2)* 是一个拉回。*3.* $(L_\mathrm{i}, u, le_\mathrm{i})$ 是由一对态射 $(k_1, \iota_K^\mathrm{m})$ 的推出计算得到的，其中$k_1 := \iota_L^{\mathrm{i}\prime} \circ le_\mathrm{b}$；在此过程中，我们选择 L_i，使得 u 和 le_i 成为单射（必要时采用重命名）。*4.* 应用条件 ac_i 通过 $Shift(\iota_L^\mathrm{i}, ac_\mathrm{b})$ 计算得到，其中 $\iota_L^\mathrm{i} := u \circ \iota_L^{\mathrm{i}\prime}$。

诱导规则 ρ_i 的大小定义为 $|\rho_\mathrm{i}| := |L'_\mathrm{i} \setminus L_\mathrm{b}| + |K_\mathrm{i} \setminus K_\mathrm{b}|$。

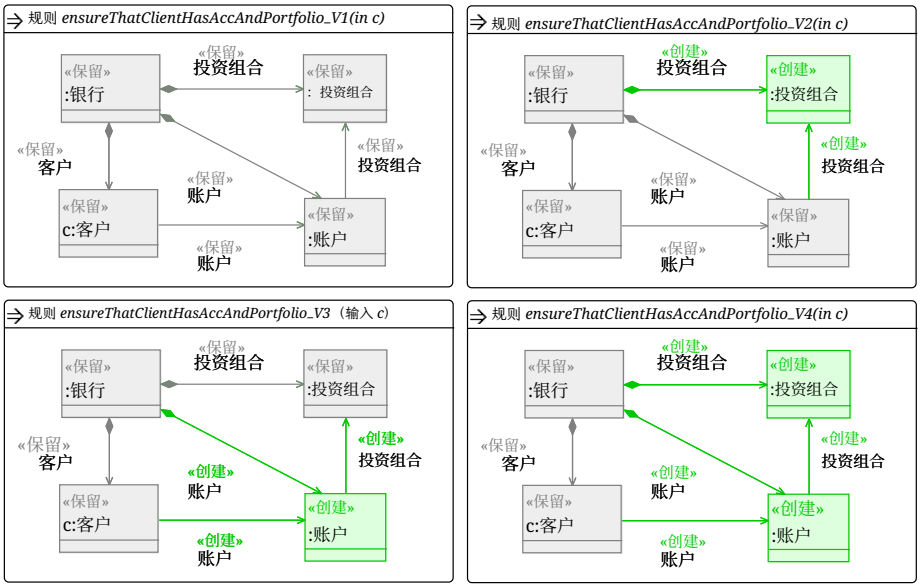

图 5。 由规则 ensureThatClientHasAccAndPortfolio 产生的四个诱导规则

示例 *2.* 再次将规则 ensureThatClientHasAccAndPortfolio 视为极大规则，并将其保留元素视为基规则。该效应导向规则有 26 个诱导规则，即所有将 «创建»元素之一重标记为ensureThatClientHasAccAndPortfolio 中的 «保留»的规则。图5展示了其中的一部分，即减少元素不期望复用的诱导规则。这些规则会在保留某个节点（而非创建该节点）的同时，也保留所有与之相邻且通向保留元素的边。我们将此称为弱连通条件，并在附录 A.1中进行讨论和形式化。）图中未展示的一个诱导规则示例是：在现有的 投资组合与 账户 和 投资组合之间创建一条新的边。

接下来，考虑这样一条效应导向规则：极大规则 ensureThat-ClientHasAccAndPortfolio 与已经创建了一个账户 及其两条入边的基规则相结合。在这里，账户 不能在任何诱导规则中成为上下文，因为其创建已经由基规则要求。（图 (2) 中的分解 4 是一个拉回，因此保证了这一点。）ensureThatClientHasNoAccAndPortfolio 的诱导规则也以类似方式获得。

我们的第一个结果表明，诱导规则实际上包含其效应导向规则的基规则作为子规则。特别是，这确保了如果应用某个诱导规则，则会执行基规则所规定的所有操作。

命题 *1*（基规则作为诱导规则的子规则）。如果 ρ_{i} 是效应导向规则 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$ 的诱导规则，那么 ρ_{b} 通过嵌入 $(\iota_L^{\mathrm{i}}, \iota_K^{\mathrm{i}}, \iota_R^{\mathrm{i}})$ 是 ρ_{i} 的子规则，其中 $\iota_L^{\mathrm{i}} := u \circ \iota_L^{\mathrm{i}\prime}$，$\iota_K^{\mathrm{i}} := \iota_K^{\mathrm{m}}$ 和 $\iota_R^{\mathrm{i}} := \iota_R$（比较图 *4*）。

接下来，我们将说明，一个效应导向规则确实能够简洁地表示一组可能很大的规则。诱导规则的确切数量取决于极大规则的边如何连接。

命题 2（诱导规则的数量）。对于一个效应导向规则 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$，其诱导子规则的数量 n （按同构计）满足：

$$2^{|V_{L_{\mathrm{m}}} \setminus V_{L_{\mathrm{b}}}| + |V_{R_{\mathrm{m}}} \setminus V_{R_{\mathrm{b}}}|} \leq n \leq 2^{|V_{L_{\mathrm{m}}} \setminus V_{L_{\mathrm{b}}}| + |E_{L_{\mathrm{m}}} \setminus E_{L_{\mathrm{b}}}| + |V_{R_{\mathrm{m}}} \setminus V_{R_{\mathrm{b}}}| + |E_{R_{\mathrm{m}}} \setminus E_{R_{\mathrm{b}}}|}.$$

尽管我们对诱导规则的定义有意设置得较为宽泛，但在许多应用场景中，限制所考虑的诱导规则的类型可能是合理的，以避免不必要的复用（例如，将一个账户连接到另一个客户已有的投资组合）。在附录A.1中，我们给出了一个能够实现这一点的定义。

4.2 匹配效应导向规则

在本节中，我们将探讨匹配效应导向规则的不同方式。效应导向变换通过所应用的效应导向规则的诱导规则，从“经典的”双推出变换中获得其语义。我们将研究不同的方式，使现有上下文能够确定应在何种匹配处应用效应导向规则的哪个诱导规则。

我们假设效应导向规则的基规则已有一个预匹配，并尝试将该预匹配扩展为适当诱导规则的匹配。相容性这一概念描述了这种扩展关系。在附录A.2中，我们提出了两个进一步刻画相容匹配的技术引理。

定义 7（相容性）。给定一个效应导向规则 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$、其基规则的预匹配 $m_{\mathrm{b}}\colon L_{\mathrm{b}} \to G$ 以及其某个诱导规则 ρ_{i} 的匹配 $m_{\mathrm{i}}\colon L_{\mathrm{i}} \to G$，若 $m_{\mathrm{i}} \circ \iota^{\mathrm{i}}_{L} = m_{\mathrm{b}}$，则二者相容；其中 $\iota^{\mathrm{i}}_{L} = u \circ \iota^{\mathrm{i}\prime}_{L}\colon L_{\mathrm{b}} \to L_{\mathrm{i}}$ 源自将 ρ_{b} 嵌入 ρ_{i} 的子规则嵌入（对比图 4 和命题 1）。

若诱导规则 ρ_{i} 存在一个匹配 m_{i}，使得 m_{b} 和 m_{i} 相容，则该规则可以与 m_{b} 相容地匹配。

给定一个效应导向规则及其基规则的预匹配，可能存在许多不同的诱导规则，使其具有相容的匹配。以下定义介绍了选择此类规则和匹配的不同策略，使相应的应用形成在删除和创建操作方面完备的变换，从而实现预期效果；这也是它们被称为效应导向变换的原因。它们的共同核心在于，在任何效应导向变换中，基规则的预匹配都会通过极大规则中的潜在创建和潜在删除加以扩展，直到无法进一步扩展为止。直观而言，这确保了所有可能的潜在删除都会执行，而仅执行必要的潜在创建（我们将在定理 1中对这一点作出精确定义）。更严格的概念是使重用元素的数量最大化。

定义 8（局部完备性、极大性、面向效果的转换）。给定一个针对面向效果规则 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$的基规则 ρ_{b} 的预匹配 $m_{\mathrm{b}}\colon L_{\mathrm{b}} \to G$，以及该规则的某个诱导规则 ρ_{i} 的匹配 m_{i}，m_{b}，且 m_{i} 相对于 m_{b} 局部完备当且仅当满足以下条件（见图 4）：

1. 额外删除的局部完备性：任何进一步的分解 $\iota_L = \iota_L^{\mathrm{m}''} \circ \iota_L^{\mathrm{i}''}$ 为单射（其定义域或陪域分别为 L_{i}''），只要存在一个非双射包含映射 j： $L_{\mathrm{i}}' \hookrightarrow L_{\mathrm{i}}''$，且满足 $j \circ \iota_L^{\mathrm{i}'} = \iota_L^{\mathrm{i}''}$ 和 $\iota_L^{\mathrm{m}''} \circ j = \iota_L^{\mathrm{m}'}$，就符合以下两个条件之一。

– 不可匹配： 不存在单射态射 e_1': $L_{\mathrm{i}}'' \hookrightarrow G$ 使得 $e_1' \circ \iota_L^{\mathrm{i}''} = m_{\mathrm{b}}$。– 不可应用：存在这样的 e_1'，但它与 m_{i} 的右扩展匹配 e_2 共同诱导的态射 m_{i}': $L_{\mathrm{i}}'' \to G$ 不是单射（其中 L_{i}'' 是与这一进一步分解相对应的诱导规则的左侧）。

2. 附加创建的局部完备性：将 $ri_{\mathrm{m}} = ri_{\mathrm{i}}'$ 。 $\iota_K^{\mathrm{m}'}$进一步分解为包含映射（定义域分别为 K_{i}'），且存在一个非双射包含映射 j： $K_{\mathrm{i}} \hookrightarrow K_{\mathrm{i}}'$，满足 $j \circ \iota_K^{\mathrm{m}} = \iota_K^{\mathrm{m}'}$，并且 ri_{i}' 。 $j = ri_{\mathrm{i}}$符合以下两个标准之一。

– 不可匹配： 不存在单射态射 e_2': $K_{\mathrm{i}}' \hookrightarrow G$，使得 $e_2' \circ \iota_K^{\mathrm{m}'} = m_{\mathrm{b}} \circ le_{\mathrm{b}}$。– 不可应用： 存在这样一个 e_2'，但它与 m_{i} 的左扩展匹配 e_1 共同诱导的态射 m_{i}': $L_{\mathrm{i}}'' \to G$ 不是单射（其中 L_{i}'' 是与这一进一步分解相对应的诱导规则的左侧）。

一个效应导向变换 t： $G \Longrightarrow H$ 经由 ρ_{e}，是一个双推出变换 t: $G \Longrightarrow_{\rho_{\mathrm{i}},\ m_{\mathrm{i}}} H$，其中 ρ_{i} 是 ρ_{e} 的诱导规则，且 ρ_{i} 关于 $m_{\mathrm{b}} := m_{\mathrm{i}} \circ \iota_L^{\mathrm{i}}$局部完备；后者是 ρ_{b}的诱导预匹配。效应导向规则的语义是其所有效应导向变换的集合。

变换 t: $G \Longrightarrow_{\rho_{\mathrm{i}},\ m_{\mathrm{i}}} H$ 经由给定效应导向规则 ρ_{e} 的诱导规则 ρ_{i}，若全局极大（关于 G），则对任意其他变换t': $G \Longrightarrow_{\rho_{\mathrm{i}}',\ m_{\mathrm{i}}'} H'$ 经由 ρ_{e}的诱导规则 ρ_{i}'，均有 $|\rho_{\mathrm{i}}| \geq |\rho_{\mathrm{i}}'|$成立。若对任意其他$_{m_{\mathrm{i}}}$变换 t': $G \Longrightarrow_{\rho_{\mathrm{i}}',\ H'}$

经由 ρ_{e} 的诱导规则 ρ_{i}'，且其针对基规则的诱导预匹配 m_{b} 和 m_{b}' 相同，均有 $|\rho_{\mathrm{i}}| \geq |\rho_{\mathrm{i}}'|$成立，则称该变换 t 为局部极大。在上述所有情况下，我们也称匹配 m_{i} 和规则 ρ_{i} 局部完备，或局部/全局极大。

示例 *3*。 为说明面向效果的规则的不同匹配类型，我们再次考虑 ensureThatClientHasAccAndPortfolio 作为一个极大规则，其«保留»-元素构成基规则，并根据不同的语义将其应用于图6所示的示例实例。首先，我们考虑将规则的 客户-节点映射到实例中的 客户 c1 的基匹配。以 局部完备 的方式扩展该基匹配，需要重用现有的投资组合 以及现有账户中的一个。选择 账户 a2 会导出规则 ensureThatClientHasAccAndPortfolio_V1 （图5），因为已经存在一条指向投资组合 p的边。相反，选择 账户 a1 会产生一个从 投资组合-边，起点为 a1，终点为 p （其底层诱导规则未在图 5中示出）。这两种变换都是局部完备的；尤其是，局部完备匹配并不具有确定性。如果出于语义上的原因，希望避免第二种变换，而只允许图中所示的诱导规则

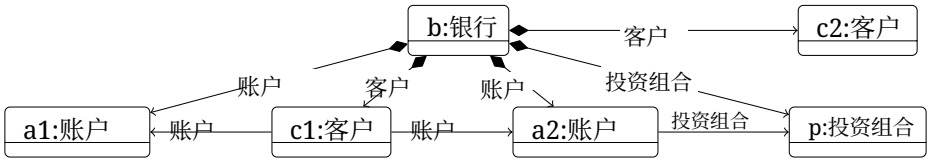

图 6. 银行领域的微型示例实例

5中的规则，则在 ensureThatClientHasAccAndPortfolio_V2 于账户 a1 处应用后，匹配就变为局部完备。

ensureThatClientHasAccAndPortfolio_V1 的唯一匹配，是本例中与所选基匹配兼容的唯一局部极大匹配，同时也是全局极大匹配。为了说明一个并非全局极大的局部极大匹配，我们考虑将 Client c2 而非 c1 映射到的基匹配。在此，局部极大匹配会重用 a2、p 以及它们之间的投资组合边，并创建从 c2 到 a2 和 p 的缺失边。若选择 Account a1 而非 a2，则无法得到局部极大匹配，因为此时无法重用投资组合边（诱导规则的大小因此减少 1）。全局极大匹配保持不变，因为根据定义，它不依赖于给定的基匹配。显然，在一个更大的示例中，每个拥有带有 Portfolio 的 Account 的 Client 都构成一个全局极大匹配。因此，全局极大匹配也是非确定性的。事实上，甚至可以构造出这样的示例：不同的诱导规则（大小相同）各自存在全局极大匹配。

与潜在创建不同，潜在删除可能需要回溯才能找到匹配。为此，考虑将规则 ensureThatClientHasNoAccAndPortfolio 视为一条极大规则，其«保留»元素构成基规则。假设一个账户可以连接多个客户，那么诱导规则的局部完备预匹配并不一定就是该规则的匹配。必须寻找一个只与已匹配客户相连的账户。

上述示例表明，已定义的各种变换概念都不是确定性的。一般而言，双推出变换也存在类似情况：匹配的选择通常具有非确定性；但在我们的情形中，同一个基匹配可能产生不同的结果。对于我们所针对的应用，例如基于规则的搜索、图修复或模型同步（参见第6.1节），这并不是问题。在这些应用中，通常只需知道，例如，应用规则后，所选的客户拥有账户和投资组合，而不一定需要知道具体是哪些账户和投资组合。

很容易看出，通过诱导规则得到的每个全局极大变换也都是局部极大变换，而每个局部极大变换都是局部完备的。因此，效果导向变换的定义涵盖了最弱的情形，也涵盖了局部极大变换和全局极大变换。

命题 3（不同类型的效应导向变换之间的关系）。给定一个效应导向规则 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$ 以及一个图 G，每个

通过 ρ_{e} 的诱导规则 ρ_{i} 得到的变换 $t\colon G \Longrightarrow_{\rho_{\mathrm{i}},\, m_{\mathrm{i}}} H$ 若为全局极大，则也是局部极大的。对于基规则 ρ_{b}，每个局部极大变换相对于其诱导预匹配 m_{b} 也是局部完备的。

直观而言，面向效果规则的极大规则规定了潜在操作的一种选择。诱导出的规则源于潜在操作的不同可能组合。下一个定理阐明了在一次效果导向变换之后最终可能发生哪些效果：如果某个元素仍然存在，而面向效果规则为它指定了一个匹配的潜在删除，那么两种情况之一已经发生：要么执行了该潜在删除，但作用于另一个元素（替代动作）——如果一个元素有多种可能的匹配方式，且可能被删除；要么该元素被某个潜在创建匹配（替代创建）。当规则为同一类型（位于可比较位置）的元素指定潜在创建和潜在删除时，就可能发生后一种情况。类似地，如果 x 表示已执行的潜在创建，但此前存在一个元素 y，而 x 本可以与之匹配，y 则被另一个潜在创建或潜在删除使用（替代动作）。特别是，定理1 还表明，面向效果的规则可以指定替代动作。其应用可能具有非确定性，即随机选择多个可能动作之一。

定理 1（效应导向变换的刻画）。设 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}},\ \rho_{\mathrm{m}},\ \iota)$ 是一个效应导向规则，而 $t\colon G \Longrightarrow_{\rho_{\mathrm{i}},\, m_{\mathrm{i}}} H$ 是通过其某个诱导规则得到的效应导向变换 ρ_{i}（下文参见图4）。

设 $x \in L_{\mathrm{m}} \mid L_{\mathrm{b}}$ 是表示 ρ_{e} 的潜在删除的一个元素，并设 $K^{+}{}_{\mathrm{b}}$ 是 K_{b} 与 x（若其定义为图）的扩展，且 $\iota^{+}\colon K_{\mathrm{b}} \hookrightarrow K^{+}{}_{\mathrm{b}}$ 是相应的包含映射。若存在一个单射态射 $m^{+}\colon K^{+}{}_{\mathrm{b}} \to H$，满足 $m^{+} \circ \iota^{+} = n_{\mathrm{i}} \circ \iota_{R} \circ \mathit{ri}_{\mathrm{b}}$，其中 n_{i} 是 t 的余匹配，则以下情况之一成立：

1.（替代动作）：元素 x 属于 L_{i}；特别地，t 从 G 中删除了一个与 x 类型相同（且位置相近）的元素；或者 *2.*（替代创建）：H 中的元素 $m^{+}(x)$ 在 n_{i} 下有一个来自 R_{i} 的原像，即它最初由 t 创建，或者与一个潜在创建匹配。

类似地，令 $x \in R_{\mathrm{i}} \mid (K_{\mathrm{i}} \cup R_{\mathrm{b}})$ 表示由 t对 ρ_{e}执行的潜在创建之一。令 $K^{+}{}_{\mathrm{b}}$ 表示通过 x 扩展 K_{b} 得到的图（如果定义为图），并令 $\iota^{+}\colon K_{\mathrm{b}} \hookrightarrow K^{+}{}_{\mathrm{b}}$ 表示相应的包含映射。则以下两种情况之一成立：

1. **(**替代动作**)**：对于每个满足 $m^{+}{}_{\mathrm{b}} \circ \iota^{+} = m_{\mathrm{i}} \circ \mathit{le}_{\mathrm{i}} \circ \iota_{K}^{\mathrm{m}}$的单射态射 $m^{+}{}_{\mathrm{b}}\colon K^{+}{}_{\mathrm{b}} \to G$，元素 $m^{+}{}_{\mathrm{b}}(x) \in G$ 都有来自 L_{i} 的原像，即在 m_{i} 下（也就是说，它已经被另一个潜在操作映射）；或 *2.* **(**匹配不存在**)**：不存在满足 $m^{+}{}_{\mathrm{b}} \circ \iota^{+} = m_{\mathrm{i}} \circ \mathit{le}_{\mathrm{i}} \circ \iota_{K}^{\mathrm{m}}$ 的单射态射 $m^{+}{}_{\mathrm{b}}\colon K^{+}{}_{\mathrm{b}} \to G$。

5 局部完备匹配——算法与实现

在本节中，我们介绍一种算法，用于从效应导向规则和一个局部完备匹配（及相应的诱导规则）中进行计算。

其基规则的预匹配。从这样的预匹配开始，非常适用于模型修复和基于规则的搜索等实际应用场景；在这些场景中，基规则的匹配通常已确定，需要通过极大规则的部分操作予以补充。请注意，所有诱导规则的所有匹配都共用此预匹配。先搜索一次预匹配，再结合上下文对其进行扩展，通常比从头开始搜索诱导规则的匹配高效得多。此外，我们同时计算局部完备匹配及其对应的诱导规则，从而避免先计算所有诱导规则（其数量可能很多，参见命题 2），再按大小顺序尝试匹配它们。我们的算法的正确性将在下文定理 2 中证明。请注意，我们着重关注算法的正确性；除上述基本的效率考量外，进一步提升效率的优化（融合 [3]中的思想）留待未来工作。我们在附录 A.3中简要说明了回溯的用法。在第 5.2节中，我们将介绍如何使用此算法在 Henshin 中实现效果导向变换的原型。

5.1 计算局部完备匹配的算法

我们考虑从给定的预匹配中寻找局部完备匹配的问题。所谓的有根规则，即部分匹配已固定（或至少可以在常数时间内确定）的规则，一直是开发在线性时间内运行的图的基于规则的算法的重要组成部分 [3,7]。请注意，我们并不是要寻找能导出极大变换的诱导规则和匹配，而只是要找到不可扩展的、即局部完备的匹配。这样一来，悬挂边条件就成为匹配过程中唯一可能导致回溯的因素。

在算法 1 中，我们概述了一个函数，该函数将效应导向规则的基规则的预匹配扩展为相应诱导规则的兼容且局部完备的匹配。我们的算法输入是一个效应导向规则$\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$（参数规则）、一个图 G （参数图）以及一个预匹配 $m_{\mathrm{b}}\colon L_{\mathrm{b}} \nrightarrow G$，用于 ρ_{b} （参数 currentMappings）。它返回一个匹配 m_{i}，用于 ρ_{e} 的诱导规则 ρ_{i}，使得 m_{i} 和 m_{b} 兼容，并且m_{i} 局部完备；当且仅当不存在这样的兼容且局部完备的匹配时，它返回 null。我们概述节点的匹配，因此将 currentMappings 视为节点映射列表；由此可以推断边的匹配。计算得到的匹配也表示相应的诱导规则。我们在附录 A.3 中详细说明了这些约定。

匹配的搜索从用给定效应导向规则的潜在操作初始化未绑定节点 开始（第 5 行）。随后，函数 *findExtension*递归地尝试匹配这些节点，并扩展预匹配（第 6 行）。函数 *find- Extension* 的工作方式如下。对于当前考虑的位置上的未绑定节点 n，使用函数 *findExtensionCandidates* 收集所有可用的候选项，即图 G 中所有可以将 n 映射到的节点 x （第 10 行）。候选项 x 必须满足以下属性：(i) 没有其他节点可以已经映射到 x，即 x 尚未出现在$currentMappings$ 中（单射条件）

以及 (ii) n 和 x 的类型必须一致（一致性条件）。如果存在候选项，算法会针对每个候选项 x尝试将 n 映射到 x，直到找到一个解。为此，当前映射集合会通过加入对（n, x）进行扩展（第 13 行）。如果 n 是最后一个待匹配的节点（第 14 行），且由$currentMappings$ 定义的态射满足悬挂边条件（第 15 行），则将结果作为解返回（第 16 行）。如果违反悬挂边条件，则移除所选候选项，并尝试下一个候选项（第 17 行）。如果 n 不是最后一个待映射的节点（第 18 行），则会针对扩展后的当前映射列表以及未绑定节点中的下一个未匹配节点调用函数 findExtension（第 19 行）。如果这会得到一个有效解，则返回该解（第 20–21 行）。否则，从 $currentMappings$ 中移除对（n, x），并尝试下一个候选项。如果存在候选项但没有任何候选项能得到有效解，则返回 null（第 33 行）。如果不存在候选项（第 25 行），则将当前映射或 null 作为解返回（如果 n 是最后一个要赋值的节点；第 26–29 行），或者调用***findExtension*** 处理下一个位置（第 31 行），也就是说，当前考虑的节点 n 会从映射中省略（因此也会从诱导规则中省略）。

算法 1. 局部完备匹配的计算

```
1 输入: 效果–导向规则（ρb, ρm, ι）、图 G 和预–匹配 mb
2 输出: 与 mb 兼容的局部完备匹配 mi
3
4 函数 findLocallyCompleteMatch(规则, 图, 当前映射)
5   未绑定节点 = V_Lm \ V_Lb ⊔ V_Rm \ V_Rb;
6   返回 findExtension(currentMappings, graph, 未绑定节点, 0);
7
8 函数 findExtension(currentMappings, 图, 未绑定节点, position)
9   n =未绑定节点.get(position);
10  候选项 = findExtensionCandidates(currentMappings, graph, n);
11  若 (! candidates.isEmpty())
12    对于候选项中的每个 x
13      currentMappings.put(n,x);
14      if (position ==unboundNodes.size() − 1) //最后一个待匹配节点
15        如果 (danglingEdgeCheck(图, currentMappings))
16          返回 currentMappings;
17        否则currentMappings.remove(n,x);
18      else //映射下一个未绑定节点
19        下一个解 = findExtension(当前映射, 图, 未绑定节点,
              位置 + 1);
20        如果 (下一个解 != 空值)
21          返回 下一个解;
22        否则//尝试下一个候选项
23            当前映射.remove(n,x);
24    结束循环 //未找到合适的候选项
25  否则 //当前节点没有候选项
26    如果 (位置 ==未绑定节点.size() − 1) //要匹配的最后一个节点
27      如果 (悬挂边检查(图, 当前匹配))
28        返回 当前匹配;
```

29 **否则返回**null; 30 **否则**//映射下一个未绑定节点31 **返回** findExtension(currentMappings, graph, unboundNodes, position+1); 32 //没有候选项能生成有效映射33 **返回**null;

定理 *2*（算法 *1* 的正确性）。给定一个效应导向规则$\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$、一个图 G以及一个预匹配 $m_{\mathrm{b}}\colon L_{\mathrm{b}}, \to G$，算法 *1* 终止并计算出一个诱导规则 ρ_{i} 及其匹配 m_{i}，使得 m_{i} 和 m_{b}兼容，并且 ρ_{i} 关于 m_{b}是局部完备的。特别地，当且仅当不存在能够与m_{b} 兼容匹配的 ρ_{e} 的诱导规则时，算法 *1* 返回 ***null***；当且仅当 m_{b} 是局部完备的且是一个匹配时，算法 *1* 返回 ρ_{b} 作为诱导规则，并返回匹配 m_{b} 。

5.2 实现

在本节中，我们介绍效应导向变换在 Henshin [1,29]，（一种基于图变换的模型变换语言）中的原型实现。从用户的角度来看，该实现包含两个主要类，用于将给定的效应导向规则 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$ 应用于宿主图 G：类 `LocallyCompleteMatchFinder` 用于查找局部完备匹配，类 `EffectOrientedRuleApplication` 则用于在此类匹配处应用规则$\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$。我们假设 $\rho_{\mathrm{e}} = (\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$ 以简单的 Henshin 规则形式给出，用来表示极大规则 ρ_{m}，其中基规则ρ_{b} 由该规则中保留的部分隐式表示。与 Henshin 中的常规做法一样，宿主图G 以给定元模型（表示类型图）的模型实例形式提供。

实现遵循第 5.1 节提出的算法。我们的主要设计目标是尽可能复用 Henshin 现有的解释器核心及其匹配和规则应用功能。具体而言，在 `LocallyCompleteMatchFinder`中，我们通过创建 ρ_{m} 的副本并移除其中的创建和删除操作，得到基规则 ρ_{b}，再将其输入解释器核心，以获得 G上的预匹配 m_{b}。对于能够找到预匹配 m_{b} 的情况，我们提供了算法 1 的一种实现，用于生成部分匹配 $\hat{m}_{\mathrm{m}}$，其中包含 m_{b} 的映射，以及针对待删除元素和不应创建元素的额外映射。为了一致地处理不同操作中的元素，我们在一条称为灰化规则的中间规则 ρ_{gr}上执行这些步骤；在该规则中，创建操作被转换为保留操作。在 `EffectOrientedRuleApplication`中，我们首先从 $\hat{m}_{\mathrm{m}}$导出诱导规则 ρ_{i}，使得 $\hat{m}_{\mathrm{m}}$ 是 ρ_{i}的完整匹配。实际应用规则时，我们将 ρ_{i} 与 $\hat{m}_{\mathrm{m}}$ 一同输入 Henshin 解释器核心，并采用经典的规则应用方式。

我们使用贯穿示例对实现进行了测试。为此，我们指定了所有规则和一个示例图。实现的运行结果完全符合预期。实现的源代码和示例可在线获取：`https://github.com/dstrueber/effect-oriented-gt`。

6 相关工作

在本节中，我们将介绍现有的实际应用如何从采用效果导向变换中获益（第 6.1 节），并将效应导向图变换与其他图变换方法进行比较（第 6.2 节）。

6.1 从效应导向图转换中获益

文献中有若干应用案例使用图变换来实现特定状态。下文将回顾图修复（目标是实现一致的图）和模型同步（目标是在其中一个模型发生更改后实现一致的模型关系）。服务匹配则是一个略有不同的案例：其目标是让现有服务的描述尽可能充分地覆盖指定服务。

在 Sandmann 和 Habel 基于规则的图修复 [28], 方法中，他们使用（可能很大的）规则集来修复形如 $\exists(a\colon B, \to C)$ 的（子）条件：对于 B' 介于 B 和 C 之间的每个图，该规则集都包含一条从 B' 创建 C 的规则。负应用条件（NAC）可确保修复过程中选用左侧具有尽可能大的 B' 的规则。使用效应导向图变换，他们推导出的整个规则集可以由一条效应导向规则表示，其基规则是 B 上的恒等变换，极大规则则为 $B, \to C$。此外，我们无需使用 NAC，因为局部完备匹配能够实现预期效果。

在模型同步的上下文中，[26] 中也会出现非常相似的情况。Orejas 等人通过模式定义模型对之间的一致性。为了进行同步，他们从单个模式推导出一整套规则，以应对恢复一致性的不同方式（即创建缺失的元素）。同样，使用 NAC 来控制规则的应用。与上文一样，我们可以将整套规则表示为单个效应导向规则。

Fritsche、Kosiol 等人扩展了基于 *TGG* 的模型同步流程，通过使用特殊的修复规则 [15,14,21]来实现更高的增量性。根据这些规则要删除的元素，可能会在同步过程中因其他原因而被删除，从而破坏修复规则所需的匹配。[15]，中通过仅考虑不会出现这种情况的编辑来避免这一问题。在 [14]，中，Fritsche 以务实的方式处理这一问题：如果需要删除的元素已经缺失，那么局部一致性便已恢复，因此可以直接跳过该删除操作。更正式地说，Kosiol 在 [21] 中提出了一组修复规则的子规则，其中总是可以选择可匹配的最大子规则来执行传播。整套规则也可以优雅地表示为单个效应导向规则。

在 [2]，Arifulina探讨了服务规范与描述的异质性。她提出了一种服务规范匹配方法：为描述服务的规则寻找最大部分匹配。除了Arifulina允许部分匹配同时省略上下文元素这一点外，这个问题

也可以表述为：在可用服务描述规则的左侧中，为服务描述规则寻找全局最大匹配（按我们的定义）。

6.2 与其他图变换方法的关系

有几种图变换方法会考虑变换上下文的可变性。下面简要回顾每种方法，并讨论它们与效应导向图变换的共性和差异。

用于应用单条变换规则的其他语义。 诸如单推出方法 [25], 拟推出方法 [10], AGREE [8], PBPO [9], 以及 PBPO$^+$ 重写 [27] 等图变换方法，比 DPO 重写表达能力更强，因为它们允许以某种方式复制或合并元素，或产生（隐式指定的）副作用。在所有这些方法中，规则都定义为（扩展）跨度。因此，它们也指定了为应用规则而必须执行的一组操作。AGREE、PBPO 和 PBPO$^+$ 可以用来指定我们所说的潜在删除；然而，在这些方法中，对它们的指定要复杂得多。上述方法都不支持指定潜在创建，而潜在创建可以根据当前考虑的应用上下文选择省略。

在双拉回方法中，“规则只规定系统状态至少必须发生什么，但允许观察可能由环境引起的其他效应” [19, 第 85] 页。在效应导向图变换中，基规则同样规定了至少必须发生什么。但附加效应并非完全任意，因为极大规则会限制附加动作。此外，只有当这些动作所规定的目标状态尚不存在时，才会执行这些附加动作。这表明，双拉回变换是比具有局部完备匹配的效应导向变换更一般的概念。

通过多条变换规则实现效应导向变换。 下面我们讨论如何利用以受控方式应用多条规则的图变换概念，来模拟效应导向变换。

图程序 [18,7]通常提供用于规则应用的控制结构，例如顺序应用、有条件应用和可选应用，以及应用循环。为了模拟效应导向图变换，首先只应用一次基规则。对于每条诱导规则，我们都会计算其余规则，即它与基规则之间的差异；换言之，该规则规定了诱导规则中基规则未规定的所有动作。为了选择正确的其余规则，我们需要一组附加规则，用于检查给定实例图中仍需执行哪些动作。根据这些检查结果，选择并应用适当的其余规则。

合并变换 [5,13] 适用于需要进行带有全称量化操作的图变换的情况。它们为交互方案提供了形式化基础：在这种方案中，内核规则恰好应用一次，额外的

多重规则扩展内核规则，并在扩展内核规则匹配的匹配处尽可能多地应用。为了通过交互方案模拟效果导向变换的行为，基本思路是生成所有可能的诱导规则，作为多重规则集合。可以使用应用条件来控制应用“正确”的诱导规则。

多个变体规则的一种紧凑表示方式是 基于变体（*VB*）的规则 [31]。VB 规则表示一组具有共同核心的规则。仅出现在部分规则中的元素会用所谓的存在条件进行注释。VB 规则可以用更复杂的方式紧凑表示效应导向规则的一组诱导规则：通过显式定义特征列表，并用这些特征标注可变部分。针对单个图 [31] 以及图的变体集合 [30]，都已定义了 VB 规则的执行语义。不过，对于 VB 规则，尚未发展出一种基于具有特定属性的匹配是否可用来驱动实例选择的概念。

7 结论

与文献中的图变换方法相比，效应导向图变换支持以更加声明式的方式对系统进行建模。基本操作的规约还伴随着对期望状态的规约。根据宿主图的不同，基规则的应用会扩展为诱导规则的应用，后者恰好执行实现期望状态所需的操作。我们讨论了效应导向变换非常适合用于规约图修复和模型同步策略，因为变更操作可以辅以相应操作，以恢复图内或多个（模型）图之间的一致性。我们概述了如何利用现有图变换方法模拟效应导向变换，但这种做法会带来效应导向变换能够避免的偶然复杂性。

未来，我们尤其感兴趣的是构造能够诱导保持一致性和改进一致性的变换的效应导向规则 [22]。研究不同方法在其匹配方面的计算复杂度，开发用于计算其匹配的高效算法，完善针对效应导向规则的冲突与依赖分析，以及将效应导向与多重融合相结合，都是具有理论和实践意义的进一步研究主题。

致谢

感谢匿名审稿人提出的建设性意见。J.K. 在伦敦国王学院担任访问研究助理期间，完成了本文部分研究工作。本研究部分得到德国研究联合会（DFG）项目资助，项目编号为 TA 294/19-1。

参考文献

1. Arendt, T., Biermann, E., Jurack, S., Krause, C., Taentzer, G.: Henshin：用于原位 EMF 模型变换的高级概念与工具。载于：Petriu, D.C., Rouquette, N., Haugen, Ø.（编）《模型驱动工程语言与系统——第13届国际会议 MODELS 2010》，挪威奥斯陆，2010年10月3—8日，论文集，第一部分。《计算机科学讲义》，第6394卷，第121–135页。施普林格（2010）。https://doi.org/10.1007/978-3-642-16145-2_9, https://doi.org/10.
1007 978 3 642 16145 2_9 / - - - - . Arifulina, S.: 为成功的服务市场解决异构性。博士论
2
文，帕德博恩大学，德国（2017 13 3 10 17619 1 ）。https://doi.org/ . /UNIPB/ - . Bak, C., Plump, D.: 根化图程序。《欧洲软件科学与技术协会电子通信》54（2012）。https://doi.org/10.14279/tuj.eceasst.54.780, https: //doi.org/10 780 4 14279 54 . / tuj.eceasst. . . Behr, N., Sobocinski, P.: 粘合范畴的规则代数。《计算机科学逻辑方法》16 2020 5 3 2020 10 23638 16 3 2 ( ) ( )。https://doi.org/ . /LMCS- (:) . Boehm, P., Fonio, H., Habel, A.: 图变换的合并及其在同步中的应用。载于：Ehrig, H., Floyd, C., Nivat, M., Thatcher, J.W.（编）《软件开发的数学基础：软件理论与实践国际联合会议（TAPSOFT）论文集》，德国柏林，1985年3月25—29日，第1卷：代数与程序设计中的树专题讨论会（CAAP'85）。《计算机科学讲义》，第185卷，第267–283页。施普林格（1985）。https://doi.org/10.1007/3-540-15198-2_17, https://doi.org/
10 1007 3 540 15198 2_17 . / - - - . Burdusel, A., Zschaler, S., John, S.:
6
自动生成用于基于搜索的模型工程、保持原子多重性的搜索算子。《软件系统建模》（2021）。https://doi.org/10.1007/s10270-021-00914-w 7. Campbell, G., Courtehoute, B., Plump, D.: 快速规则驱动的图程序。《科学计算编程》214，102727（2022）。https://doi.org/10.1016/j.scico.2021.102727, https://doi.org/10 102727 8 1016 2021 . / j.scico. . . Corradini, A., Duval, D., Echahed, R., Prost, F., Ribeiro, L.: AGREE——带受控嵌入的代数图重写。载于：Parisi-Presicce, F., Westfechtel, B.（编）《图变换——第8届国际会议 ICGT 2015》，作为 STAF 2015 的一部分举行，意大利拉奎拉，2015年7月21—23日，论文集。《计算机科学讲义》，第9151卷，第35–51页。施普林格（2015）。https://doi.org/10.
1007 978 3 319 21145 9_3 10 1007 978 3 319 21145 9_3 / - - - - ,
9
https://doi.org/ . / - - - - . Corradini, A., Duval, D., Echahed, R., Prost, F., Ribeiro, L.: PBPO 图变换方法。《逻辑、代数方法与程序学报》103，213–231（2019）。https://doi.org/10.1016/j.jlamp.2018.12.003, https://doi.org/10.1016/j.jlamp.2018. 12.003 10. Corradini, A., Heindel, T., Hermann, F., König, B.: 半推出重写。载于：Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., Rozenberg, G.（编）《图变换：第3届国际会议 ICGT 2006》，巴西北里奥格兰德州纳塔尔，2006年9月17—23日，论文集。《计算机科学讲义》，第4178卷，第30–45页。施普林格（2006）。https://doi.org/10.1007/11841883_4, https://doi.org/ 10 1007 11841883_4 . / . Corradini, A., Montanari,
11
U., Rossi, F., Ehrig, H., Heckel, R., Löwe, M.: 图变换的代数方法——第一部分：基本概念与双推出方法。载于：Rozenberg, G.（编）《图文法与图变换计算手册》，第1卷：基础，第163–246页。世界科学出版社（1997）

12. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.:《代数图转换基础》。理论计算机科学专著。EATCS丛书，施普林格（2006）。https://doi.org/10.1007/3-540-31188-2, https://doi.org/10.1007/ 3-540-31188-2 13. Ehrig, H., Golas, U., Habel, A., Lambers, L., Orejas, F.: 带嵌套应用条件的M-粘合变换系统。第1部分：并行性、并发性与合并。《数学结构与计算机科学》24(4)（2014）。https://doi.org/10.1017/ S0960129512000357, https://doi.org/10.1017/S0960129512000357 14. Fritsche, L.: 三图文法的局部一致性恢复方法。博士论文，达姆施塔特工业大学，德国（2022），http://tuprints. ulb.tu-darmstadt.de/21443/ 15. Fritsche, L., Kosiol, J., Schürr, A., Taentzer, G.: 避免不必要的信息丢失：基于三图文法的正确且高效的模型同步。《国际软件工具与技术转移期刊》23(3), 335–368（2021）。https://doi.org/10. 1007/s10009-020-00588-7, https://doi.org/10.1007/s10009-020-00588-7 16. Golas, U., Habel, A., Ehrig, H.: m-粘合范畴中带应用条件的规则多重合并。《数学结构与计算机科学》24(4)（2014）。https://doi.org/10.1017/S0960129512000345 17. Habel, A., Pennemann, K.: 相对于嵌套条件的高级变换系统正确性。《数学结构与计算机科学》19(2), 245–296（2009）。https: //doi.org/10.1017/S0960129508007202 18. Habel, A., Plump, D.: 基于图变换的编程语言的计算完备性。载于：Honsell, F., Miculan, M.（编）《软件科学与计算结构基础》，第四届国际会议 FOSSACS 2001，作为软件理论与实践联合欧洲会议 ETAPS 2001 的一部分于2001年4月2日至6日在意大利热那亚举行，会议论文集。《计算机科学讲义》第2030卷，第230–245页。施普林格（2001）。https: //doi.org/10.1007/3-540-45315-6_15, https://doi.org/10.1007/3-540-45315-6_1519. Heckel, R., Ehrig, H., Wolter, U., Corradini, A.: 图变换系统的双拉回转移与余代数宽松语义。《应用范畴结构》9(1), 83–110（2001）。https://doi.org/10.1023/A:1008734426504 20. Horcas, J.M., Strüber, D., Burdusel, A., Martinez, J., Zschaler, S.: 我们不会把它搞砸！用于高效产品线配置的保持一致性的算子。《IEEE软件工程汇刊》（2022）。https://doi.org/10.1109/TSE. 2022.3171404 21. Kosiol, J.: 基于三图文法的信息保持型模型同步过程的形式化基础。博士论文，马尔堡大学，德国（2022），https://archiv.ub.uni-marburg.de/diss/z2022/0224 22. Kosiol, J., Strüber, D., Taentzer, G., Zschaler, S.: 维持并提升渐进式图一致性：图变换的静态分析。《计算机编程科学》214（2021）23. Krause, C.: 银行账户示例。在线（2023），https://wiki.eclipse.org/ Henshin/Examples/Bank_Accounts 24. Lack, S., Sobociński, P.: 粘合范畴与准粘合范畴。《RAIRO理论信息学与应用》39(3), 511–545（2005）。https://doi.org/10.1051/ita:2005028, https://doi.org/10.1051/ita:2005028 25. Löwe, M.: 单推送图转换的代数方法。《理论计算机科学》109(1&2), 181–224（1993）。https://doi.org/10.1016/0304-3975(93) 90068-5, https://doi.org/10.1016/0304-3975(93)90068-5

26. Orejas, F., Guerra, E., de Lara, J., Ehrig, H.: 基于模式的模型到模型变换的正确性、完备性与终止性。载于：Kurz, A., Lenisa, M., Tarlecki, A.（编）《计算机科学中的代数与余代数》，第三届国际会议 CALCO 2009，2009年9月7日至10日在意大利乌迪内举行，会议论文集。《计算机科学讲义》第5728卷，第383–397页。施普林格（2009）。https://doi.org/10.1007/978-3-642-03741-2_26，https://doi.org/10.1007/978-3-642-03741-2_2627. Overbeek, R., Endrullis, J., Rosset, A.: 使用pbpo进行图重写与重新标记+。载于：Gadducci, F., Kehrer, T.（编）《图变换——第14届国际会议》，ICGT 2021，25，2021，2021，24 12741，60 80，作为STAF虚拟活动的一部分于6月举行，会议论文集。《计算机科学讲义》第卷，第–页。施普林格（2021）。https://doi.org/10.1007/978-3-030-78946-6_4，https://doi.org/10.1007 978 3 030 78946 6_4 /----。Sandmann, C., Habel, A.: 基于规则的图修复。载于：28 Echahed, R., Plump, D.（编）第十届图计算模型国际研讨会论文集，GCM@STAF 2019，2019年7月17日于荷兰埃因霍温举行。EPTCS，第309卷，第87–104页（2019）。https://doi.org/10.4204/EPTCS.309.5，https://doi.org/10.4204/ EPTCS.309.529. Strüber, D., Born, K., Gill, K.D., Groner, R., Kehrer, T., Ohrndorf, M., Tichy, M.: Henshin：用于EMF模型变换开发的面向可用性的框架。载于：ICGT'17：图变换国际会议。第196–208页。施普林格，查姆（2017）。https://doi.org/10.1007/978-3-319-61470-0_12，https:// doi.org/10 1007 978 3 319 61470 0_12 。/----。Strüber, D., Peldszus, S., Jürjens, J.: 驾驭30 软件产品线变换的多变性。载于：FASE。第337–355页（2018）。https://doi.org/10.1007/978 3 319 89363 1_19 10 1007 978 3 319 89363 1_19 ----，31 https://doi.org/./----。Strüber, D., Rubin, J., Arendt, T., Chechik, M., Taentzer, G., Plöger, J.: 基于多变性的模型变换：形式化基础与应用。《计算机形式化方面》**30**(1), 133–162（2018）。https://doi.org/10.1007/s00165-017-0441-3

A 补充说明与结果

A.1 诱导规则的连通性条件

如例 2所示，效应导向规则的一些诱导规则在语义上可能值得商榷，例如将某个客户 的账户 连接到某个任意的、已存在的投资组合 (该投资组合可能属于另一位客户)。我们对诱导规则的定义有意保持宽泛，以免排除在某些应用上下文中可能有用的规则。特别是，按照 Sandmann 和 Habel 的建议，通过效应导向规则实现图修复 [28] (参见第6.1节)需要采用这种宽泛的定义。然而，对于某些应用，进一步限制所考虑的诱导规则类型可能更合理。我们对连通性条件的定义展示了如何做到这一点。

定义 *9*（连通性条件）。给定一个效应导向规则 ρ_{e}，若其诱导规则满足以下条件，则称其满足弱左连通条件：在图 *4* 中的因式分解 *(1)* 里，对于每个节点 $x \in V_{L'_{\mathrm{i}}} | \iota'^{\mathrm{i}}_{L}(V_{L_{\mathrm{b}}})$，如果存在一条边 $e \in E_{L_{\mathrm{m}}}$，使得 $src_{L_{\mathrm{m}}}(e) = \iota'^{\mathrm{m}}_{L}(x)$ 且 $tar_{L_{\mathrm{m}}}(e) \in \iota'^{\mathrm{m}}_{L}(V_{L'_{\mathrm{i}}})$，或 $tar_{L_{\mathrm{m}}}(e) = \iota'^{\mathrm{m}}_{L}(x)$ 且 $src_{L_{\mathrm{m}}}(e) \in \iota'^{\mathrm{m}}_{L}(V_{L'_{\mathrm{i}}})$，那么该边也属于 L'_{i} （即在

$\iota_L^{\mathrm{m}\prime}$下有原像）。类似地，若对于每个节点 $x \in V_{K_\mathrm{i}} \mid \iota_K^{\mathrm{m}}(V_{K_\mathrm{b}})$，如果存在一条边 $e \in E_{R_\mathrm{m}}$，使得 $src^{R_\mathrm{m}}(e) = ri_\mathrm{i}(x)$ 且 $tar^{R_\mathrm{m}}(e) \in ri_\mathrm{i}(V_{K_\mathrm{i}})$，或 $tar^{R_\mathrm{m}}(e) = ri_\mathrm{i}(x)$ 且 $src^{R_\mathrm{m}}(e) \in ri_\mathrm{i}(V_{K_\mathrm{i}})$，那么该边也属于 K_i（即在 ri_i 下有原像），则称其满足弱右连通条件。

若对于每个节点 $x \in V_{L_\mathrm{i}'} \mid \iota_L^{\mathrm{i}\prime}(V_{L_\mathrm{b}})$ $(x \in V_{K_\mathrm{i}} \mid \iota_K^{\mathrm{m}}(V_{K_\mathrm{b}})$，每条相邻边 $e \in E_{L_\mathrm{m}}$ $(e \in E_{R_\mathrm{m}})$ 也属于 L_i' (K_i)，则该诱导规则满足左连通性条件（右连通性条件）。

示例 4。示例2 中的诱导规则包括允许连接已有账户与投资组合，或账户与客户的规则，也就是说，这些规则允许任意复用元素。图5展示了满足弱右连通性条件的诱导规则。虽然这些规则仍允许一定程度上可能不受欢迎的复用（例如，将现有的投资组合连接到新的账户），但可能的复用已受到很大限制。例如，已无法在现有元素之间建立新的连接。

在我们的示例中，唯一满足右连通性条件的诱导规则是图 ensureThatClientHasAccAndPortfolio V1 和ensureThat-_ClientHasAccAndPortfolio_ V4 中的规则。5这是因为在此示例中，所有可能创建的节点都通过边相连。

A.2 进一步刻画相容匹配

接下来的两个引理将进一步刻画相容匹配。第一个结果使我们能够从基规则的预匹配出发，通过逐步纳入各个潜在删除或创建的映射，为诱导规则构造匹配。

引理 1（相容性的刻画）。给定一个效应导向规则$\rho_\mathrm{e} = (\rho_\mathrm{b}, \rho_\mathrm{m}, \iota)$ 及其基规则的预匹配 m_b，当且仅当满足以下条件时，诱导规则 ρ_i 的ρ_e 可与 m_b 相容匹配（参见图 4）：

1. 左侧兼容性：存在一个单射态射 $e_1\colon L_\mathrm{i}' \rightarrowtail G$，称为左扩展匹配，使得 $e_1 \circ \iota_L^{\mathrm{i}\prime} = m_\mathrm{b}$。 *2.* 右侧兼容性：存在一个单射态射 $e_2\colon K_\mathrm{i} \rightarrowtail G$，称为右扩展匹配，使得 $e_2 \circ \iota_K^{\mathrm{m}} = m_\mathrm{b} \circ \mathit{le}_\mathrm{b}$。 *3.* 适用性：由推出的泛性质可知存在唯一态射 $m_\mathrm{i}\colon L_\mathrm{i} \to G$，且 $e_2 \circ \iota_K^{\mathrm{m}} = m_\mathrm{b} \circ \mathit{le}_\mathrm{b} = e_1 \circ \iota_L^{\mathrm{i}\prime} \circ \mathit{le}_\mathrm{b} = e_1 \circ k_1$ 是单射，并满足 ρ_i 的悬挂条件。

在此情形下，唯一的态射 m_i 是满足 m_b 和 m_i兼容的 ρ_i 的匹配。

证明。 首先，若 ρ_i 满足所列要求，根据第三项要求，我们得到单射态射 $m_\mathrm{i}\colon L_\mathrm{i} \rightarrowtail G$，它是 ρ_i的预匹配。根据假设，m_i 满足 ρ_i的悬挂条件。此外，$m_\mathrm{b} \models ac_\mathrm{b}$ 意味着 $m_\mathrm{i} \models ac_\mathrm{i} = \mathrm{Shift}(\iota_L^\mathrm{i}, ac_\mathrm{b})$，依据 [13, 引理 3.11]。因此，$m_\mathrm{i}$ 是 ρ_i的匹配。最后，

根据其泛性质，该态射满足 $m_{\mathrm{i}} \circ u = e_1$，这意味着$m_{\mathrm{i}} \circ \iota_L^{\mathrm{i}} = m_{\mathrm{i}} \circ u \circ \iota_L^{\mathrm{i}'} = e_1 \circ \iota_L^{\mathrm{i}'} = m_{\mathrm{b}}$; 也就是说， m_{b} 与 m_{i} 相容。

反过来，给定一个诱导规则 ρ_{i} 的匹配 m_{i}，且满足$m_{\mathrm{i}} \circ \iota_L^{\mathrm{i}} = m_{\mathrm{b}}$，也就是说，它与 m_{b} 相容，我们可以定义 $e_1 := m_{\mathrm{i}} \circ u$ 和 $e_2 := m_{\mathrm{i}} \circ le_{\mathrm{i}}$。这意味着

$$\begin{aligned} e_1 \circ \iota_L^{\mathrm{i}'} &= m_{\mathrm{i}} \circ u \circ \iota_L^{\mathrm{i}'} \\ &= m_{\mathrm{i}} \circ \iota_L^{\mathrm{i}} \\ &= m_{\mathrm{b}} \end{aligned}$$

and

$$\begin{aligned} e_2 \circ \iota_K^{\mathrm{m}} &= m_{\mathrm{i}} \circ le_{\mathrm{i}} \circ \iota_K^{\mathrm{m}} \\ &= m_{\mathrm{i}} \circ u \circ k_1 \\ &= m_{\mathrm{i}} \circ u \circ \iota_L^{\mathrm{i}'} \circ le_{\mathrm{b}} \\ &= m_{\mathrm{b}} \circ le_{\mathrm{b}}, \end{aligned}$$

也就是说， ρ_{i} 满足左侧和右侧兼容性。此外，将上述推导中的第三个等式替换后，这一推导还表明 $e_2 \circ \iota_K^{\mathrm{m}} = e_1 \circ k_1$。因此， m_{i} 是唯一的态射，其存在性由计算 L_{i} 的推出的泛性质所蕴含。根据假设， m_{i} 是一个匹配，因此它是单射，并且满足 ρ_{i} 的悬挂条件。由此， ρ_{i} 满足全部三个要求。证毕

接下来，我们刻画左侧兼容性和右侧兼容性与上述引理中适用条件之间的关系。特别是，这使我们能够论证，在算法 1 中，只有对潜在删除的匹配才会导致回溯。

引理 2（确保适用性）。给定一个效应导向规则 $\rho_{\mathrm{e}} =(\rho_{\mathrm{b}}, \rho_{\mathrm{m}}, \iota)$，以及其基规则的预匹配 $m_{\mathrm{b}}\colon L_{\mathrm{b}} \nrightarrow G$，和针对某个诱导规则的左扩展匹配 $e_1\colon L_{\mathrm{i}}' \nrightarrow G$ 与右扩展匹配 $e_2\colon K_{\mathrm{i}} \nrightarrow G$，分别满足左侧和右侧兼容性条件。则诱导的唯一态射$m_{\mathrm{i}}\colon L_{\mathrm{i}} \to G$ 是单射，且满足 ρ_{i} 的悬挂边条件，当且仅当 ***(i)*** e_1 和 e_2 只识别原本就源自 $K_{\mathrm{b}} = K_{\mathrm{m}}$的元素，即若 $e_1(x) = e_2(y)$，则元素 x 和 y 在 K_{b}中有公共原像；并且 ***(ii)*** e_1 满足态射 k_1的悬挂边条件，即若

$$\{v \in V_{L_{\mathrm{i}}'} \mid \exists e \in E_G \setminus e_1(E_{L_{\mathrm{i}}'}).src_G(e) = e_1(v) \text{ or } tar_G(e) = e_1(v)\} \subseteq k_1(V_{K_{\mathrm{b}}}). \quad (1)$$

证明。 我们分别讨论单射性和悬挂边条件；关于以下证明，也请参见图 4。

如果 m_{i} 是单射，则两个态射 $e_1 = m_{\mathrm{i}} \circ u$ 和 $e_2 = m_{\mathrm{i}} \circ le_{\mathrm{i}}$ 作为单射态射的复合也是单射。此外，由于计算L_{i} 的推出（在粘合范畴中）同时也是拉回，且 m_{i} 是单射，因此外部方形 $e_1 \circ k_1 = e_2 \circ \iota_K^{\mathrm{m}}$也是拉回。这意味着 $e_1(L_{\mathrm{i}}') \cap e_2(K_{\mathrm{i}}) = K_{\mathrm{b}}$。

另一方面，如果 $e_1(L_i') \cap e_2(K_i) = K_b$，这意味着外部方形$e_1 \circ k_1 = e_2 \circ \iota_K^m$ 是拉回。于是，根据 [24, 定理 5。1]， m_i 是单射。

关于适用性，引理中的等式表达了悬挂边条件（参见 [12, 定义 3.9 和事实 3.11]）。我们需要证明

$$M_1 := \{v \in V_{L_i'} \mid \exists e \in E_G \setminus e_1(E_{L_i'}).\, src_G(e) = e_1(v) \text{ or } tar_G(e) = e_1(v)\} \subseteq k_1(V_{K_b}) \tag{2}$$

当且仅当

$$M_2 := \{v \in V_{L_i} \mid \exists e \in E_G \setminus e_1(E_{L_i}).\, src_G(e) = m_i(v) \text{ or } tar_G(e) = m_i(v)\} \subseteq le_i(V_{K_i}). \tag{3}$$

假设等式 2 成立，令 $v \in M_2$。如果 $v \notin \mathsf{le}_i(V_{K_i})$，则存在 $v' \in V_{L_i'}$，使得 $u(v') = v$（因为 u 和 le_i 作为一个推出的余投影是联合满射的）。根据 $e_1 = m_i \circ u$，这意味着 $v' \in M_1$；特别地，$v' \in k_1(V_{K_b})$。这与 $v \notin \mathsf{le}_i(V_{K_i})$ 矛盾。因此，$M_2 \subseteq \mathsf{le}_i(V_{K_i})$。

反过来，假设等式 3 成立，令 $v \in M_1$。如果 $u(v) \in M_2$，则等式 3 蕴含 $u(v) \in \mathsf{le}_i(V_{K_i})$，进而由于计算 L_i 的推出同时也是一个拉回，推出 $v \in k_1(V_{K_b})$。如果 $u(v) \notin M_2$，则存在一条边 $e' \in E_{L_i}$，满足 $m_i(e') = e$，其中 $e \in E_G$ 是导致 $v \in M_1$ 的边之一。由于 u 和 le_i 联合满射，e' 在其中一个映射下有原像。因为 $e' \in u(E_{L_i'})$ 会与 e 导致 $v \in M_1$ 这一事实矛盾，所以 $e' \in \mathsf{le}_i(E_{K_i})$。这也导致 $u(v) \in \mathsf{le}_i(V_{K_i})$（因为 K_i 是一个图，而 le_i 是一个同态），进而如上所述，蕴含 $v \in k_1(V_{K_b})$。综上，等式 3 蕴含 $M_1 \subseteq k_1(V_{K_b})$。证毕

A.3 匹配过程讨论

下文将解释节点映射列表如何同时表示边的映射以及底层的诱导规则，并简要说明算法1中为何需要回溯。

在算法 1 中，相关数据具有以下形式：预匹配m_b 由已分配节点对的列表 currentMappings 给出。该列表以下述方式表示匹配 m_i 和左侧 L_i（或其当前状态）：L_i 的节点是 n 和 L_m 中被分配给节点 x的节点，其中的分配由 currentMappings 中的节点对 (n, x) 指定。这种分配也定义了节点上的 m_i。L_i 的边是 L_m 和 R_m 中能够最大限度纳入的边，使得 m_i 可以在这些边上扩展为到 G的单射态射。

此约定存在两种可能的歧义来源：首先，在 L_m、R_m 和/或 G 中，相同两个节点之间可能存在多个同类型的平行边。在这种情况下，所有可能的选择都会导出同构规则，删除/创建的该类型边的数量也相同。其次，可能存在一对源自 K_b 的节点，使得 ρ_m 同时规定删除并创建它们之间某种类型的边。在这种情况下，该类型边的潜在创建和删除会在匹配时争夺相同的边。不过，每种匹配和诱导规则的选择都会导出变换的同构结果。我们也可以排除此类规则，因为它们不具有语义意义（它们规定在同一位置删除并重新创建同一种边）。最后，对于每一对 $(n,\ x)$，位于

当前映射，我们知道节点 n 源自 $V_{L_\mathrm{m}} \setminus V_{K_\mathrm{b}}$ 或源自V_{R_m} （包括 V_{K_b}）。在第一种情况下， n 是要删除的节点；在第二种情况下， n 是要由诱导规则保留的节点。因此，要获得相应的诱导规则，只需将 R_m中缺失的元素添加为创建。

请注意，算法 1 之所以需要回溯，仅仅是因为要删除节点。如果不存在可能被删除的节点，那么根据 $V_{L'_\mathrm{i}} = V_{L_\mathrm{b}}$，左扩展匹配满足公式 1 （因此，根据引理2，它也会为其诱导规则导出一个匹配）当且仅当它所扩展的预匹配 m_b 本身已经是基规则的匹配。因此，在这种情况下，我们只需对 m_b检查一次悬挂边条件。如果不满足该条件，我们就返回 null；否则，算法中无论选择哪个候选项，都会得到一个匹配，并且不会发生回溯。具体而言，可以将计算所有可能候选项（第 10 行）替换为搜索单个候选项。这样，在这一特定情况下即可高效地计算局部完备匹配。

B 主要结果的证明

证明（命题 1）。以下证明可参见图7。其中，方框 (2) 根据假设是一个拉回。此外，左上方的方框也是一个拉回，因为$k_1 = \iota_L \circ \mathsf{le}_\mathrm{b}$ （根据假设），且 ι_L 是单射的（例如，参见 [4, 引理 1.10.(III)]）。左下方的方框是沿单射态射作推出，因此也是一个拉回（例如，参见[12, 注 2.24]）。根据拉回复合（例如，参见 [12, 事实 2.27]），左侧两个方框合起来构成一个拉回。总之，规则 ρ_b 通过拉回嵌入到 ρ_i 中，正如所需。最后，$\mathsf{ac}_\mathrm{i} = \mathsf{Shift}(\iota_L, \mathsf{ac}_\mathrm{b})$由定义可得。证毕。

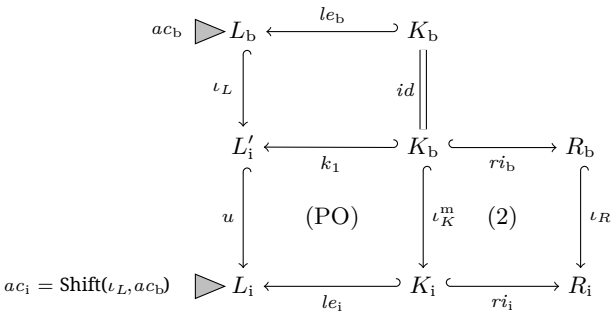

图7. 证明 ρ_b 是 ρ_i 的子规则

证明（命题 2）。任意一对节点集合 $(A,\ B)$，其中 $A \subseteq V_{L_\mathrm{m}} \setminus V_{L_\mathrm{b}}$且 $B \subseteq V_{R_\mathrm{m}} \setminus V_{R_\mathrm{b}}$，都会导出一条诱导规则。这是通过令 L'_i （见图 4）为扩展 L_b 并加入来自 A 的节点所得的图，并令 K_i 为扩展 $K_\mathrm{b} = K_\mathrm{m}$ 并加入来自 B的节点所得的图；注意， B 中的节点具有

源自 $V_{R_\mathrm{m}} \setminus V_{R_\mathrm{b}}$（而非源自 $V_{R_\mathrm{m}} \setminus V_{K_\mathrm{m}}$），以确保 (2) 是一个拉回。这样的节点集合对有 $2^{|V_{L_\mathrm{m}} \setminus V_{L_\mathrm{b}}| + |V_{R_\mathrm{m}} \setminus V_{R_\mathrm{b}}|}$ 个。模同构意义下，这些也是在节点上定义 L'_i 和 K_i 的所有可能方式。对于边而言，如果来自 $E_{L_\mathrm{m}} \setminus E_{L_\mathrm{b}}$ 或 $E_{R_\mathrm{m}} \setminus E_{R_\mathrm{b}}$ 的边可以独立于节点的选择而纳入其中，那么 L'_i 和 K_i 的选择就有最大的灵活性。（这恰好等价于每条这样的边都关联到已经分别源自 L_b 或 R_b 的节点。）在这种情况下，模同构意义下，构造 L'_i 和 K_i 对的方法有 $2^{|V_{L_\mathrm{m}} \setminus V_{L_\mathrm{b}}| + |E_{L_\mathrm{m}} \setminus E_{L_\mathrm{b}}| + |V_{R_\mathrm{m}} \setminus V_{R_\mathrm{b}}| + |E_{R_\mathrm{m}} \setminus E_{R_\mathrm{b}}|}$ 种。证毕

证明（命题 *3*）。显然，全局极大变换也是局部极大的。关于第二个陈述，若某条规则不是局部完备的，那么它就不可能是局部极大的，因为现有的扩展会导出一个规模更大的可应用诱导规则。证毕

证明（定理 1）。首先，令 $x \in L_\mathrm{m} \setminus L_\mathrm{b}$ 为表示 ρ_e 的一个潜在删除的元素；若其可定义为一个图，令 $K^+\,\mathsf{b}$ 为 K_b 与 x 的扩展，并令 $\iota^+\colon K_\mathrm{b} \hookrightarrow K^+\,\mathsf{b}$ 为相应的包含映射。假设 $m^+\colon K^+\,\mathsf{b} \to H$ 是一个单射态射，满足 $m^+ \circ \iota^+ = n_\mathrm{i} \circ \iota_R \circ \mathsf{ri}_\mathrm{b}$，其中 n_i 是给定效果导向变换 t的余匹配。首先，可能存在这样一种情况：$m^+(x)$ 在 n_i 下有来自 R_i 的原像 y。（这意味着 t 要么先创建了$m^+(x)$——即 $y \in R_\mathrm{i} \setminus K_\mathrm{i}$ 的情况；要么 y 是潜在创建——即 $y \in K_\mathrm{i} \setminus K_\mathrm{b}$ 的情况。）

若 $m^+(x)$ 在 n_i下没有来自 R_i 的原像，则存在一个元素 $y \in G$，使得变换 t 的所谓跟踪态射将 y 映射到 $m^+(x)$（即 $h(g^{-1}(y)) = m^+(x)$；参见图 2）。设 $\iota'^{\mathrm{i}}_L\colon L_\mathrm{b} \hookrightarrow L'_\mathrm{i}$ 源自 ι_L 的分解，该分解用于计算诱导规则 ρ_i，并假设 $x \notin L_\mathrm{i}$（因此也有 $x \notin L'_\mathrm{i}$）。在这种情况下，我们可以通过用 x 扩展 L'_i 来定义 L^+_i，并通过将映射 $x \mapsto y$加入左扩展态射 e_1 来扩展 $L^+_\mathrm{i} \hookrightarrow G$，从而定义 e'_1。条件 $m^+ \circ \iota^+ = n_\mathrm{i} \circ \iota_R \circ \mathsf{ri}_\mathrm{b}$，加上 $m^+(x)$ 经 t 保持且在 n_i下没有原像这一事实，确保所得的是一个单射态射。因此，计算 ρ_i 的 ι_L 的分解违反了不可匹配条件。此外，引理 2 表明它也违反了不可应用条件，因为在这种情况下，y 在右扩展匹配 e_2 下没有原像。总之，假设 $x \notin L_\mathrm{i}$ 会导致矛盾，因此 $x \in L_\mathrm{i}$。（直观地说，这意味着变换 t 删除的是另一个可能的元素，而不是 y。）

其次，令 $x \in R_\mathrm{i} \setminus (K_\mathrm{i} \cup R_\mathrm{b})$ 表示变换 t已执行的 ρ_e的某个潜在创建。令 $K^+\,\mathsf{b}$ 为带有 x 的 K_b的扩展（若将其定义为图），K'_b 为 $K^+\,\mathsf{b}$ 与 K_i的交集，并令 $\iota'\colon K'_\mathrm{b} \hookrightarrow K_\mathrm{i}$和 $\iota^+\colon K'_\mathrm{b} \hookrightarrow K^+\,\mathsf{b}$ 为相应的包含映射。假设存在一个单射态射 $m^+\,\mathsf{b}\colon K^+\,\mathsf{b} \to G$，使得 $m^+\,\mathsf{b} \circ \iota^+ = m_\mathrm{i} \circ \mathsf{le}_\mathrm{i} \circ \iota^{\mathrm{m}}_K$。按照与上述过程完全类似的步骤，我们可以通过将 x 纳入 K_i，扩展由 ρ_i的计算得出的 $\iota_R \circ \mathsf{ri}_\mathrm{b}$ 的分解；我们还将 e_2扩展，将 x 映射到 G中的 $m^+\,\mathsf{b}\,(x)$。若所得态射不是单射，则$m^+\,\mathsf{b}\,(x)$ 在 e_2下有另一个原像；特别地，它在 m_i下也有一个原像。

若所得态射是单射，则计算 $\iota_R \circ ri_{\mathrm{b}}$ 的 ρ_{i} 分解违反了不可匹配条件。由于 t 根据假设是一个效果导向变换，这意味着该分解满足不可应用条件。同样，根据引理2，这意味着 $m^{+}{}_{\mathrm{b}}(x)$ 在 e_1 下有一个原像。特别地，在这种情况下，$m^{+}{}_{\mathrm{b}}(x)$ 在 m_{i} 下也有一个原像。（直观而言，这意味着 $m^{+}{}_{\mathrm{b}}(x)$ 已经被其他某个潜在操作——删除或创建——匹配。）证毕

（定理 *2* 的证明）。算法1 的整体终止性*findExtension*函数的终止性所保证。在此函数中，*for* 循环遍历一个有限的候选项集合（第 12–24 行），且在每种情况下，回溯深度都受列表未绑定节点的长度限制（在第 14 行和第 26 行检查）。因此，*findExtension*会终止。

关于正确性，首先，findExtension 会扩展 currentMappings，但始终不会更改该列表的开头，其中存储着基规则的预匹配 m_{b}。因此，如果算法 1 返回一个匹配，按构造它便与 m_{b} 兼容（假设 findExtensionCandidates（在第 10 行调用）的正确性成立）。此外，findExtension 实现了用于深度优先搜索的标准递归方案。它会尝试所有满足悬挂边条件的匹配方式。因此，当且仅当不存在能与 m_{b}兼容匹配的 ρ_{e} 诱导规则时，findExtension（以及 findLocallyCompleteMatch）才会返回 null。否则，它会计算出一个匹配。最后，所计算出的匹配是局部完备的，因为在 findExtension 中，只要存在合适的候选项，潜在操作总会被映射：只有在没有可供映射的候选项时，才会跳过潜在操作的匹配（第 11 行和第 25 行）；如果不存在任何能使某个给定节点的映射产生匹配的可能性，findExtension 就会返回 null（第 33 行）。这意味着，在 findExtension 执行期间，如果 unboundNodes 中位置 k 处的节点 n 不属于 currentMappings，那么 findExtension 要么当前是针对位置 $\leq k$ 调用的，要么无法通过为 n 赋值来进一步扩展位置 1 至 $k-1$ 处节点的赋值，否则就会违反额外删除的局部完备性条件（定义 8）。ut