面向字符串图重写的临界对枚举算法*

AnnaMatsui Innocent Obi约翰斯·霍普金斯大学，美国 华盛顿大学，美国
amatsui1@jhu.edu innoobi@cs.washington.eduGuillaumeSabbagh Leo Torres贡皮˙涅技术大学，法国† 国立科˙尔多瓦大学，阿根廷 guillaume.sabbagh@utc.fr leo.torres@mi.unc.edu.arDiana Kessler JuanF. Meleiro塔林理工大学，爱沙尼亚‡ 圣˜保罗大学，巴西 diana-maria.kessler@taltech.ee juan.meleiro@usp.br

Koko Muroya日本国立信息学研究所，日本 & 御茶水女子大学，日本§kmuroya@is.ocha.ac.jp

临界对分析为多种重写系统提供了一种方便且可计算的合流性判据，而合流性是重写理论中的一项基本性质。Bonchi 等人证明了临界对分析对于对称幺半范畴中字符串图上的重写是有效的。本研究旨在实现字符串图重写的临界对分析自动化，并开发一种实现临界对分析核心部分的算法。该算法枚举给定左连通字符串图重写系统的所有临界对，并且可以通过对超图进行具体操作来实现。我们证明了该算法对于不含 Frobenius 结构的对称幺半范畴中的字符串图是正确且完备的。

1 引言

1.1 重写理论与临界对分析

数学推理通常涉及从已知（通常较简单的）方程推导出（复杂的）方程，这有时称为等式推理。方程可以涉及各种数学对象，例如项、程序、图、过程，以及范畴中的对象/态射。

重写理论已经建立（例如参见 [2, 15]），等式推理是其一种应用。起点是将已知等式 $a = a'$ 转化为有向的[1] 重写规则 $a \multimap a'$。每一步 $b \to b'$ 重写，都是通过应用一条重写规则来修改 b 的一部分。要推导等式 $c ? = c'$，归根结底就是找到某个 d，使得存在两条重写链 $c \to \cdots \to d$ 和 $c' \to \cdots \to d$。这两条链合起来蕴含一条等式链 $c = \cdots = d = \cdots = c'$，从而得出所需的等式 $c ? = c'$。

*我们感谢伴随学校 2024 的组织者，我们的合作正是从那里开始的。
†G.S. 由法国巴黎银行企业及投资银行部欧洲、中东和非洲区以及法国研究部通过 CIFRE 项目（编号 2021/1502）资助。
‡D.K. 由先进研究 + 发明署（ARIA）安全保障人工智能：TA1.1 理论资助项目资助。§K.M. 由日本学术振兴会、科研费项目编号 22K17850（日本）资助。
[1]方向的选取通常使得 a' 比 a“更简单”。

A. Hadzihasanovic 与 J.-S. P.Lemay：应用范畴论 2025（ACT 2025）EPTCS 442，2026，第 91–105 页，
doi:10.4204/EPTCS.442.7

© A. Matsui 等人。本作品采用知识共享署名许可协议授权。

合流性是重写理论中的一项基本性质，直观而言，它意味着重写的顺序无关紧要。若任意两条分岔的重写链 $b \leftarrow \cdots \leftarrow a \rightarrow \cdots \rightarrow b'$都可汇合，则称重写 $\rightarrow$具有合流性；也就是说，存在 c，使得两条重写链 $b \rightarrow \cdots \rightarrow c \leftarrow \cdots \leftarrow b'$汇聚。局部合流性是合流性的一种变体，其中两条分岔链实际上分别由一次重写构成，即 $b \leftarrow a \rightarrow b'$。

对于某些两个发散重写 $b \leftarrow a \rightarrow b'$ 的对，其可合流性是显而易见的。一 个例子是所谓的并行情形，即两个重写修改了 a 中不同且相互独立的部分。因此，检查局部合流性归结为分析非并行对的可合流性。

临界对分析是一种成熟的自动检查局部合流性的技术，提供了一项便捷且可计算的判据。它将局部合流性归结为临界对的可合流性；临界对是非并行对的有限个代表。给定一组重写规则 $a \multimap a'$，即可枚举出临界对。该枚举在自动化局部合流性检查中发挥着核心作用。

1.2 字符串图重写

字符串图 [12, 16] 提供了范畴论的图形语法。它们对于范畴中态射的等式推理十分有用，因为它们将某些等式简化为图同构。Bonchi 等人 [3, 4, 5], 已经发展了字符串图的重写理论，旨在研究对称幺半范畴中的字符串图（包括具有和不具有 Frobenius 结构的情形）。字符串图使用超图进行组合表示，而字符串图上的重写则通过双推出重写（简称 DPO 重写）在范畴论上建模 [7]。字符串图重写理论中的一个关键概念是接口。超图的接口规定了其他超图如何与该超图连接。

Bonchi 等人证明了临界对分析对于字符串图重写的有效性 [5]。他们针对一种将接口纳入考量的 DPO 重写变体（称为凸[2] DPOI 重写）定义了临界对，并证明临界对的可合流性意味着局部合流性。他们的工作侧重于理论层面，而自动化作为临界对分析的重要方面，尚未得到研究。

1.3 贡献

我们的目标是实现字符串图重写中临界对分析的自动化，并开发一种实现自动化核心部分的算法。该算法枚举给定 DPOI 重写规则集的所有临界对。我们聚焦于所谓的左连通 DPOI 重写规则 [4, 5]。左连通性使我们能够将临界对的枚举归约为超图范畴中某些余跨的枚举。尽管这是一项颇具限制性的条件，但它仍涵盖了文献中各种具体的字符串图重写系统 [9, 10, 13]。

每个临界对都与两条 DPOI 重写规则相关联，它们由超图范畴中的跨度 $L_1 \leftarrow K_1 \rightarrow R_1$ 和 $L_2 \leftarrow K_2 \rightarrow R_2$ 给出。得益于左连通性，该临界对由某个形如 $L_1 + L_2 \twoheadrightarrow S \leftarrow J$ 的余跨唯一确定。特别是，它的左腿是由单态射的耦合给出的满态射。

我们的核心思想是，余跨，特别是超图 S，可以通过适当地粘合 $L_1 + L_2$ 的超边和节点来生成（即将 L_1 与 L_2 并行放置的超图）。我们观察到，粘合过程可以分两步实现：（1）反复将 L_1 中的一条超边与 L_2, 中的一条超边合并；（2）反复将 L_1 中的一个节点与 L_2 中的一个节点合并，而不合并任何超边。

[2]凸性用于处理缺乏 Frobenius 结构的情况。

我们的贡献可总结如下。

- 我们通过实现两重粘合过程，开发了一种算法（算法3），用于枚举给定左连通 DPOI 重写规则集的所有临界对。
- 我们证明，该算法生成所有且仅生成临界对（正确性与穷尽性；定理 3.9）。

- 我们提供了一个概念验证性质的 Haskell 实现[3]。
- 我们提出了一种优化算法（算法4），该算法只执行两重粘合过程的第一步，便能枚举数量较少但足以判定局部合流性的临界对。

组织结构。第 2 节回顾弦图重写理论中的相关概念（例如超图、接口、DPOI 重写和临界对）[3, 4, 5]。第 3 节介绍我们的主要贡献，即临界对枚举算法及其正确性与穷尽性的证明。第 4 节介绍优化算法，第 5 节总结全文。示例和部分证明见附录。

相关工作。对于项重写，重写规则通常使用变量作为占位符，以简洁地表示一族重写规则，例如 $x+y \multimap y+x$。为处理变量，临界对的枚举会采用一种称为合一的技术。相比之下，对于字符串图重写，重写规则始终是具体的，不含占位符。因此，我们可以采用直接的方法，通过适当地粘合重写规则左部中的超边和节点来生成临界对。已有一些工作尝试为图重写（图变换）的变体枚举临界对，例如 [14, 6, 11]。

2 字符串图重写的临界对

我们将态射 $f: A \to B$ 和 $g: B \to C$ 的复合记为 $f;g$，并将余积的余投影记为 ι_1,ι_2。给定集合 A，A 上的自由幺半群记为 A^*。函数 $f: A \to B$ 对元素列表逐点作用，得到 $f^*: A^* \to B^*$。令 $\mathbb{N}$ 为自然数集。

2.1 带接口的超图

当一个对称幺半范畴配备了 Frobenius 结构时，该范畴中的字符串图可以组合地表示为带接口 [3] 的（边标记）超图。

定义 2.1（超图）。一个（有向）超图是一个元组 $G = (V, E, s: E \to V^*, t: E \to V^*)$，其中 V 和 E 分别是节点和超边的有限集合；s 将每条超边映射到一个源节点列表，t 将每条超边映射到一个目标节点列表。超边的元数是其源的数量，超边的余元数是其目标的数量。

我们将 V 记作 Nodes(G)，将 E 记作 HEdges(G)。

设 σ 是一个字母表，Σ 上的签名 σ 是 $\sigma \times \mathbb{N} \times \mathbb{N}$ 的子集。三元组 (x, n, m) 表示一个用于元数为 n、余元数为 m 的态射的标签。一个带 Σ 标签的超图（简称 Σ-超图）是一个配备标记函数 $l: E \to \Sigma$ 的超图，其中 l_E 将一条元数为 n、余元数为 m 的超边映射到三元组 (x,n,m)。

定义 2.2（超图态射）。两个 Σ-超图 $(V_0, E_0, s_0, t_0, l_0)$ 与 $(V_1, E_1, s_1, t_1, l_1)$ 之间的 Σ-超图态射是一对函数 $f_V: V_0 \to V_1$ 与 $f_E: E_0 \to E_1$，它们保持源、目标和标签；也就是说，满足 $f^*_V \circ s_0 = s_1 \circ f_E$、$f^*_V \circ t_0 = t_1 \circ f_E$ 以及 $l_0 = l_1 \circ f_E$。

[3] 可在 https://github.com/GuiSab/hypergraphrewriting 在线获取

给定签名 Σ， Σ-超图与 Σ-超图态射构成一个范畴 Hypᴇ。它具有所有小极限和余极限（因为它是一个预层范畴 [3, 第 18-19] 页）；特别地，它具有推出、余积和余等化子。我们可以用集合论项把它们具体写出来，从而使其适合算法实现。

接口指定了超图中可与其他超图相连的节点。

定义 2.3（带接口的 Σ-超图）。离散 Σ-超图是不含超边的 Σ-超图（即当 HEdges(G) 为空集时，G 是离散的，它只包含节点）。带接口的 Σ-超图是 Hypᴇ 中的余跨 $n \longrightarrow G \longleftarrow m$，其中 n、m 是有限离散 Σ-超图。

离散超图 n 和 m 分别指定输入接口和输出接口。我们有时用单个态射 $G \leftarrow n+m$ 来表示带接口的 Σ-超图 $n \longrightarrow G \longleftarrow m$。对于不具备 Frobenius 结构的一般对称幺半范畴，字符串图的组合表示需要对超图 [4]施加额外条件：单配性和无环性。

定义 2.4（路径）。 Σ-超图中的路径 P 是超边列表 $[e_1, e2, \cdots, e_n]$，使得对于每一对连续超边 (e_k, e_{k+1})，e_k 至少有一个目标等于 e_{k+1} 的一个源。 Σ- 超图中的环 C 是一条路径，使得 e_1 至少有一个源是 e_n 的一个目标。

定义 2.5（单配无环性）。一个 Σ-超图是单配无环的（ma-超图），如果 1. 它不包含环（无环性）；

2. 每个节点的入度和出度至多为 1（单配性）。

这里，Σ-超图 H 中节点 v 的入度（出度）是满足以下条件的对 (e,i) 的数量：e 是 H 的一条超边，且 v 是其第 i 个目标（源）。我们将入度为 0 的节点称为输入节点，记为 in(H)。类似地，出度为 0 的节点为输出节点，记为 out(H)。

一个带有接口 $n \xrightarrow{f} H \xleftarrow{g} m$ 的 Σ-超图是单配无环的，或称 ma-余跨图，当且仅当 H 是 ma-超图，f 是单态射且其像包含于 in(H) 中，并且 g 是单态射且其像包含于 out(H) 中。

2.2 凸 DPOI 重写

字符串图上的重写可以通过调整 DPO 重写 [7] 以范畴论方式建模 [3, 4], 。我们首先回顾 Hypᴇ 中的 DPO 重写。

重写规则是 Hypᴇ 中的一个跨度 $L \longleftarrow K \longrightarrow R$。重写系统 $\mathscr{R}$ 是一个有限的重写规则集。如果存在一条规则 $L \leftarrow K \rightarrow R$、一个态射 $L \xrightarrow{m} G$（称为匹配）以及一个对象 $C \in \mathrm{Hyp}_\Sigma$，使得下列两个方块是推出，则称 Σ-超图 G 重写为 Σ-超图 H:

$$\begin{array}{ccccc} L & \xleftarrow{f} & K & \xrightarrow{g} & R \\ {\scriptstyle m}\downarrow & \ulcorner & \downarrow & & \downarrow \\ G & \longleftarrow & C & \longrightarrow & H \end{array}$$

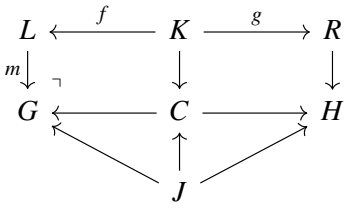
上述重写的过程如下。计算推出补会从 G 中移除 L 的像（即重写规则的左侧），同时保留 K 的像。通过计算 $C \longleftarrow K \xrightarrow{g} R$ 的推出，我们沿着 K 的像将 R 与 C 粘合，从而用 R 替换 G 中 L 的像。更直观地说，这一过程会移除与重写规则左侧 L 的匹配相对应的部分，并用其右侧 R 替代。

在本文中，我们重点研究左连通重写规则。

定义 2.6（强连通性）。如果对于每个输入 $x \in \mathrm{in}(G)$ 和输出 $y \in \mathrm{out}(G)$，G 中都存在一条从 x 到 y 的路径，则称 ma-超图 G 为强连通。

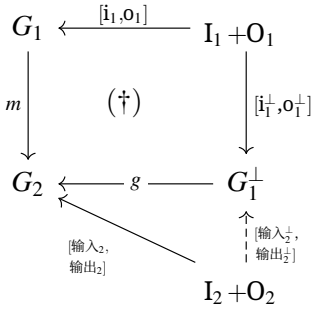
左连通重写规则是一个跨度 $L \xleftarrow{[L\ \ L]}{}_{i,\ o}I + O \xrightarrow{[R\ \ R]}{}_{i,\ o}R$，使得 $I \xrightarrow{i\ L} L \xleftarrow{o\ L} O$ 和 $I \xrightarrow{i\ R} R \xleftarrow{o\ R} O$ 是 ma-余跨度，$[i_L、o_L]$ 是单态射（我们称该规则为左线性），且 L 是强连通的。
左连通重写系统是一组左连通重写规则。
将 DPO 重写应用于字符串图的第一项改编，是使其能够处理接口。这由此产生 DPOI 重写 [3]。给定两个带接口的超图 $G \leftarrow J$ 和 $H \leftarrow J$，如果存在一个重写规则 $L \xleftarrow{f} K \xrightarrow{g} R$、一个匹配 $L \xrightarrow{m} G$，以及一个带接口的超图 $C \leftarrow J$，使得下方的方块是推出且整个图表可交换，则称 G 重写为 H：

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "\urcorner", very near start] & K \arrow[l, "f"'] \arrow[r, "g"] \arrow[d] & R \arrow[d] \\
G & C \arrow[l] \arrow[r] & H \\
 & J \arrow[ul] \arrow[u] \arrow[ur] &
\end{tikzcd}$$



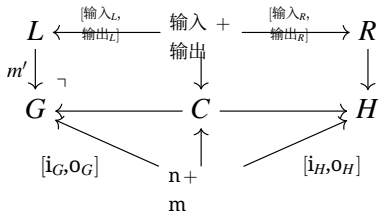
对 DPO 重写的第二项改编，是对匹配施加凸性条件。这是为了处理缺少 Frobenius 结构 [4] 的情况所必需的。
定义 2.7（凸匹配）。一个 Σ-超图态射 $m : L \to G$ 若为单态射，且其像 m(L) 是凸的，则称其为凸匹配；也就是说，对于 m(L) 中任意节点 v、v′ in，以及 G 中从 v 到 v′ in 的任意路径 p，p 中的每条超边也都属于 m(L)。
我们回顾 [4, 定义 30] 中边界补的定义。
定义 2.8（边界补）。设 $I_1 \xrightarrow{i\ 1} G_1 \xleftarrow{o\ 1} O_1$ 和 $I_2 \xrightarrow{i\ 2} G_2 \xleftarrow{o\ 1} O_2$ 是两个 ma-余跨图，且 $m : G_1 \to G_2$ 是一个单态射。如下方 (†) 所示的推出补是边界补，当且仅当 $[i_1^\perp、o_1^\perp]$ 是单态射，并且存在 $i_1^\perp : I_2 \to G_1^\perp$ 和 $o_1^\perp : O_2 \to G_1^\perp$，使得下方三角形交换，且 $O_1 + I_2 \xrightarrow{[o_1^\perp\ \ {}_2^\perp],i} G_1^\perp \xleftarrow{[i_1^\perp,o\ {}_2^\perp]} I_1 + O_2$ 是一个 ma-余跨图。

$$\begin{tikzcd}
G_1 \arrow[d, "m"'] \arrow[dr, phantom, "(\dagger)"] & I_1 + O_1 \arrow[l, "{[i_1,o_1]}"'] \arrow[d, "{[i_1^\perp,o_1^\perp]}"] \\
G_2 & G_1^\perp \arrow[l, "g"'] \\
 & I_2 + O_2 \arrow[ul, "{[\text{输入}_2,\ \text{输出}_2]}"] \arrow[u, dashed, "{[\text{输入}_2^\perp,\ \text{输出}_2^\perp]}"']
\end{tikzcd}$$



定义 2.9（凸重写）。给定一个左连通重写系统 $\mathscr{R}$，若存在一个凸匹配 $m' : L \to G$、一个重写规则 $L \xleftarrow{[L\ \ L]}{}_{i,o}I + O \xrightarrow{[R\ \ R]}{}_{i,o}R$（属于 $\mathscr{R}$）以及一个 Σ-超图 C，使得下图交换、左方形是边界补且右方形是推出，则称 ma-余跨图 $n \xrightarrow{i\ G} G \xleftarrow{o\ G} m$ 凸重写为 $n \xrightarrow{i\ H} H \xleftarrow{o\ H} m$：

$$\begin{tikzcd}
L \arrow[d, "m'"'] \arrow[dr, phantom, "\urcorner", very near start] & \text{输入} + \text{输出} \arrow[l, "{[\text{输入}_L,\ \text{输出}_L]}"'] \arrow[r, "{[\text{输入}_R,\ \text{输出}_R]}"] \arrow[d] & R \arrow[d] \\
G & C \arrow[l] \arrow[r] & H \\
 & n + m \arrow[ul, "{[i_G,o_G]}"] \arrow[u] \arrow[ur, "{[i_H,o_H]}"'] &
\end{tikzcd}$$



我们将 $\mathrm{n} \overset{G}{\underset{i}{\rightarrow}} \mathrm{G} \overset{G}{\underset{o}{\leftarrow}} \mathrm{m} \Rightarrow_{\mathscr{R}} \mathrm{n} \overset{H}{\underset{i}{\rightarrow}} \mathrm{H} \overset{H}{\underset{o}{\leftarrow}} \mathrm{m}$ 记作推导。

得益于左连通性，一个推导可以由一个 ma-余跨图 $\mathrm{n} \rightarrow \mathrm{G} \leftarrow \mathrm{m}$、一个单态匹配 $\mathrm{L} \overset{'}{\underset{m}{\rightarrow}} \mathrm{G}$ 和一条重写规则 $\mathrm{L} \leftarrow \mathrm{I} \rightarrow \mathrm{O} \rightarrow \mathrm{R}$ 唯一确定。

命题 2.10。在左连通重写系统中，边界补条件总是满足的。在左连通重写系统中，单态匹配总是凸的。

命题 2.11。在左连通重写系统中，对于所有重写规则和所有单态匹配，推出补 C 总是唯一存在的。

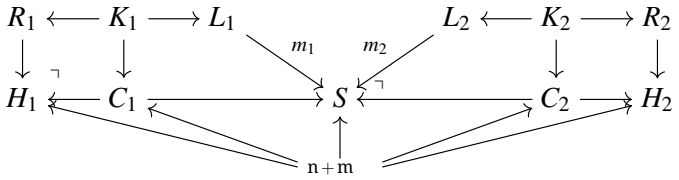
证明。存在性由 [8, pp.44,45] 得出，而推出补的唯一性由 [3,命题 3.18] 得出。

□

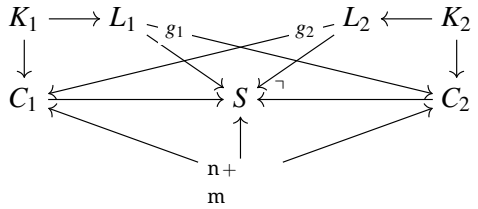
2.3 临界对

最后，我们回顾临界对的定义 [5]。

定义 2.12（临界对）。设 $\mathscr{R}$ 是一个左连通重写系统，$\mathrm{L}_1 \leftarrow\!\!\shortmid \mathrm{K}_1 \rightarrow\!\!\shortmid \mathrm{R}_1$ 和 $\mathrm{L}_2 \leftarrow\!\!\shortmid \mathrm{K}_2 \rightarrow\!\!\shortmid \mathrm{R}_2$ 是其中的两条重写规则。考虑两个具有公共源 $\mathrm{n} \rightarrow \mathrm{S} \leftarrow \mathrm{m}$ 的推导：



1. 若 $[\mathrm{m}_1,\mathrm{m}_2]: \mathrm{L}_1 + \mathrm{L}_2 \rightarrow \mathrm{S}$ 是满态射，则称 $(\mathrm{n} \rightarrow \mathrm{H}_1 \leftarrow \mathrm{m}) \Leftarrow_{\mathscr{R}} (\mathrm{n} \rightarrow \mathrm{S} \leftarrow \mathrm{m}) \Rightarrow_{\mathscr{R}} (\mathrm{n} \rightarrow \mathrm{H}_2 \leftarrow \mathrm{m})$ 是一个预临界对。
2. 若存在 W 使得 $(\mathrm{n} \rightarrow \mathrm{H}_1 \leftarrow \mathrm{m}) \overset{*}{\Rightarrow}_{\mathscr{R}} (\mathrm{n} \rightarrow \mathrm{W} \leftarrow \mathrm{m}) \overset{*}{\Leftarrow}_{\mathscr{R}} {}_{*}(\mathrm{n} \rightarrow \mathrm{H}_2 \leftarrow \mathrm{m})$，其中 $\Rightarrow_{\mathscr{R}}$ 表示有限次（可能为零次）重写。
3. 如果存在 $\mathrm{g}_1 : \mathrm{L}_1 \rightarrow \mathrm{C}_2$ 和 $\mathrm{g}_2 : \mathrm{L}_2 \rightarrow \mathrm{C}_1$，使下图表交换，则该预临界对是平行对：



4. 如果该预临界对不是平行对，则它是非临界对。

根据命题 2.10 和命题 2.11，对于左连通重写系统，预临界对 $(\mathrm{n} \rightarrow \mathrm{H}_1 \leftarrow \mathrm{m}) \Leftarrow_{\mathscr{R}} (\mathrm{n} \rightarrow \mathrm{S} \leftarrow \mathrm{m}) \Rightarrow_{\mathscr{R}} (\mathrm{n} \rightarrow \mathrm{H}_2 \leftarrow \mathrm{m})$ 可以由一个余跨（称为 cp-余跨）$\mathrm{L}_1 + \mathrm{L}_2 \twoheadrightarrow \mathrm{S} \leftarrow \mathrm{I} \rightarrow \mathrm{O}$ 唯一确定，该余跨满足：(i) $\mathrm{L}_1 + \mathrm{L}_2 \twoheadrightarrow \mathrm{S}$ 是一个满态射，由两个从 L_1 和 L_2 到 S 的单态匹配耦合而成；(ii) $\mathrm{I} \rightarrow \mathrm{S} \leftarrow \mathrm{O}$ 是一个 ma-余跨图。

3 临界对枚举算法

我们的目标是自动枚举给定左连通重写系统的所有临界对。为此，我们必须枚举所有相关的满态射 $L_i + L_j \twoheadrightarrow S$，其中 L_i 和 L_j 是重写规则的左部。我们将粘合 $L_i + L_j$ 的不同节点和超边，以枚举这些满态射。

定义 3.1. 设 L_1 与 L_2 为两个 Σ-超图。粘合方案由一个 Σ-超图 G 以及两个 Σ-超图态射 $g_1 : G \to L_1 + L_2$ 与 $g_2 : G \to L_1 + L_2$ 给出。粘合是 g_1 与 g_2 的余等化子。对于 $L_1 + L_2$, 的两个节点（或超边）x 与 x′，若存在 G 的一个节点（或一条超边）y，使得 $g_1(y) = x$ 且 $g_2(y) = x'$，则它们被粘合。

$$G \underset{g_2}{\overset{g_1}{\rightrightarrows}} L_1 + L_2 \xrightarrow[\varepsilon]{} \mathsf{coeq}_G(g_1, g_2)$$

每个粘合方案 $(g_1 : G \to L_1 + L_2, g_2 : G \to L_1 + L_2)$ 都诱导一个余跨：

$$L_1 + L_2 \twoheadrightarrow \mathsf{coeq}_G(g_1, g_2) \xleftarrow{[\subseteq,\subseteq]} in(\mathsf{coeq}_G(g_1, g_2)) + out(\mathsf{coeq}_G(g_1, g_2))$$

我们将余等化子 $\mathsf{coeq}_G(g_1, g_2)$ 称为候选源。如第 2.3 节所述，若该余跨是 cp-余跨，则它唯一确定一个预临界对。现在的问题是：粘合方案需要满足哪些必要条件，才能使该余跨成为 cp-余跨，从而诱导一个（预）临界对？我们首先观察到，粘合方案不应在 L_1 与 L_2 内部粘合节点或超边。

命题 3.2. 若将 L_1（分别为 L_2）中的两个节点粘合，则该粘合方案不会产生预临界对。若将 L_1（分别为 L_2）中的两个超边粘合，则该粘合方案不会产生预临界对。

证明。若在 $\mathsf{coeq}_G(g_1, g_2)$ 中将 L_1 中的两个节点粘合，则 $\iota_1;\varepsilon$ 不是单态射，因此不是有效的凸匹配。超边的情况证明相同。 □

其次，我们注意到，来自 L_1 和 L_2, 的节点应以特定方式分别进行粘合。

命题 3.3. 若将 L_1 中的节点 A 与 L_2 中的节点 B 粘合，且该粘合方案产生预临界对，则满足以下条件之一：

- A 和 B 是某个粘合超边的第 k 个源；
- A 和 B 是某个粘合超边的第 i 个目标；
- A 是 L_1 的输出，B 是 L_2 的输入。
- A 是 L_1 的输入，B 是 L_2 的输出；

命题 3.4。如果来自 L_1 的节点 A 和来自 L_2 的节点 B 被粘合，且粘合方案产生一个预临界对，则：

- 如果 A 是 L_1 的输出，B 是 L_2 的输入，则 L_1 的任何输入都不会与 L_2 的输出粘合；
- 如果 A 是 L_1 的输入，B 是 L_2 的输出，则 L_1 的任何输出都不会与 L_2 的输入粘合。

证明。假设来自 L_1 的节点 A 和来自 L_2 的节点 B 被粘合，粘合方案产生一个预临界对，A 是 L_1，的输出，B 是 L_2 的输入。假设 L_1 的输入 X 与 L_2 的输出 Y 粘合。由 L_1 和 L_2 的强连通性可知，存在从 X 到 A 以及从 B 到 Y 的路径。于是，我们得到一个环 $[X] \rightsquigarrow [A] = [B] \rightsquigarrow [Y] = [X]$，这与无环性质矛盾。

第二点也可以通过类似的论证得出。 □

最后，我们可以 获得粘合方案诱导出临界对的充分必要条件 .

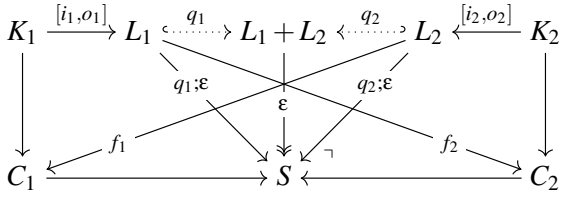
命题 3.5. 一个预临界对

$$L_1 + L_2 \twoheadrightarrow \mathtt{coeq}_G(g_1, g_2) \xleftarrow{[\subseteq,\subseteq]} in(\mathtt{coeq}_G(g_1, g_2)) + out(\mathtt{coeq}_G(g_1, g_2))$$

由粘合方案产生的平行对，当且仅当满足以下条件：

1. L_1 和L_2 中没有超边被粘合；2. 如果L_1 和L_2 中的两个节点被粘合，
那么它们分别属于L_1 和L_2的接口。

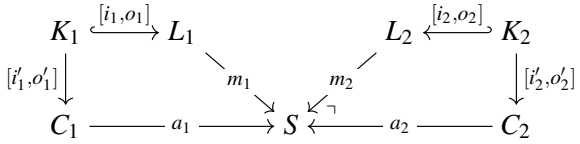
⇒的证明。假设我们有一个平行对



• 设$v_1 \in L1$、$v_2 \in L2$是两个被粘合的节点（根据命题 3.2，它们必定来自不同的超图）。由于这是一个平行对，存在映射$f1 : L_2 \to C1$、$f2 : L_1 \to C2$，使得上图可交换。由于三角形可交换，f1(v2)必须被映射到 $[v_2] = [v_1] \in S$。此外，$(q1;\varepsilon)(v1) = [v_1] = [v_2] \in S$。但S是一个推出，因此C1和L1中被识别的元素必须存在于K1中，所以v1和f1(v2)在K1中有原像，且 $v_1 \in [i1、o_1](K1)$。由对称性，同理可得$v_2 \in [i2、o_2](K2)$。

• 假设存在粘合超边。设 $e_1 \in L1$ 和 $e_2 \in L2$ 映射到同一个 $e \in S$。与之前一样，通过图表追踪可知，被识别的节点属于接口 $[i1、o_1], [i2、o_2]$ 。但 K1、K2 是离散的，因此这类超边不可能存在——不存在 $e'_1 \in C2$、$e'_2 \in C1$，使得 $f_2^{-1}(e'_1) = e1$ 且 $f_1^{-1}(e'_2) = e2$，并且图表可交换。

⇐的证明。假设条件 1 和 2 成立，下面证明该预临界对是平行对。考虑



证明通过图表追踪完成。

如果没有节点被粘合，那么也没有超边被粘合，并且 $S \cong L_1 + L_2$ 显然是一个平行对。否则，考虑一对节点 $(v_1 \in L_1, v_2 \in L_2)$，它们在 S 中具有相同的像 $v = m_1(v_1) = m_2(v_2)$。根据假设，v_2 属于 $[i_2,o_2] : K_2 \to L_2$的像。令 $f_2 : L_1 \to C_2$ 为将 v_1 映射到$[i'_2、o'_2](v_2)$，并在其余节点和边上保持恒等的态射（因为根据假设，L_1 和 L_2 中的边没有被粘合）。

根据构造，这将使三角形交换。

对于 $f_1 : L_2 \to C_1$，同样如此。 □

命题 3.6。一个预临界对

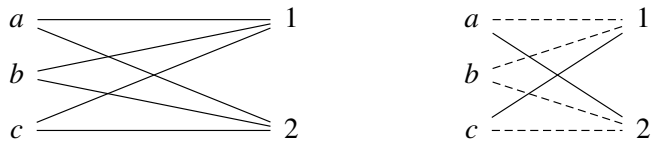
$$L_1 + L_2 \twoheadrightarrow \mathtt{coeq}_G(g_1, g_2) \xleftarrow{[\subseteq,\subseteq]} in(\mathtt{coeq}_G(g_1, g_2)) + out(\mathtt{coeq}_G(g_1, g_2))$$

由粘合方案产生的结果是临界对，当且仅当 L_1 和 L_2 中分别有超边被粘合。

证明。这是命题 3.5 的推论，且 $\mathrm{coeq}_G(g_1, g_2)$ 是一个 ma-超图。 . □

这些观察结果表明，可采用以下两重粘合过程来得到一个合适的粘合方案（$g_1 : G \to L_1 + L_2, g_2 : G \to L_1 + L_2$），从而产生一个临界对：(1) 粘合分别来自 L_1 和 L_2, 的（至少一对）超边；(2) 粘合分别来自 L_1 和 L_2的输入/输出。
计算　这样的粘合方案时，我们使用完全二分图上的独立边集 s.

定义 3.7. 给定两个集合 A 和 B，完全二分图 $K_{A,B}$ 定义如下：其顶点为 A +B，并且 A 中的每个元素与 B 中的每个元素之间都有一条边。
$K_{A,B}$ 上的独立边集[4] 是一个边集，其中任意两条边都没有共同顶点。
以下展示了一个独立边集的例子。



$K_{\{a,b,c\},\ \{1,2\}}$

对于超边上的每个独立边集，即 $K_{HEdges(L_1),\ 超边(L_2)}$（或者对于节点上的每个独立边集，即 $K_{Nodes(L_1),\ 节点(L_2)}$），我们可以如下构造诱导超图 γ：对于独立边集中的每一条连接两个顶点的边，我们都添加与这两个顶点相关联的超边对（或节点对），以及它们各自诱导的源和目标对。令 $p_1^\gamma : \gamma \to L_1$ 为 γ 到 L_1 的第一投影，令 $p_2^\gamma : \gamma \to L_2$ 为 γ 到 L_2 的第二投影。因此，我们得到一个粘合方案 $(p_1^\gamma; \iota_1 : \gamma \to L_1 + L_2, p_2^\gamma; \iota_2 : \gamma \to L_1 + L_2)$。
通过首先在超边上适当地生成独立边集（即 $K_{HEdges(L_1),超边(L_2)}$），其次在节点上（即 $K_{Nodes(L_1),节点(L_2)}$），尤其是其对输入和输出的限制，我们可以计算出粘合方案，这些方案按照独立边集所指定的方式粘合超边和节点，从而导出一个临界对；如果二分图的两个顶点之间存在一条边，则该边的两个端点会被合并（粘合）。

我们的算法使用子程序来枚举完全二分图 $K_{a,\ b}$上的独立边集。我们假设集合 a 和 b 是全序的。我们认为这是一个合理的假设，因为超图的超边集和节点集通常使用全序数据结构实现。zip 函数将两个长度相同的列表转换为一个配对列表。

算法 1：enumerateIndependentEdgeSets，用于枚举独立边集

输入：集合 a 和集合 b
输出：$K_{a,b}$ 上的独立边集
1 返回 $\bigcup_{k \in [\![0, \min(|a|, |b|)]\!]}$ enumerateKIndependentEdgeSets(a, b, k) ;

[4]独立边集在图论中也称为“匹配”。

算法2：用于枚举 k 个独立边集的 enumerateKIndependentEdgeSets

输入：集合 a、集合 b 和一个数 k
输出：$K_{a,b}$ 上具有 k 条边的独立边集

```
对于满足 |x| = k 的 x ⊆a，执行
    对于满足 |y| =k 的 y′ ⊆b 的部分排列 y，执行
        yield zip(x,y)
    end
结束
```

命题3.8。$K_{a,\ b}$上有 $\sum_{0\le k\le min(|a|,|b|)} k!\binom{|a|}{k}\binom{|b|}{k}$ 个独立边集。

现在我们可以给出临界对枚举算法（算法3）。隐式定义的子程序 InducedHypergraphs 用于计算给定超边或节点上的独立边集所对应的诱导超图。

下面的定理验证了算法3。特别是，正确性意味着命题 3.2、命题3.3 和命题3.4 中确定的必要条件同时也是充分条件。

定理3.9。算法3是正确且穷尽的。也就是说，

Correctness *Each result* $L_i + L_j \overset{\varepsilon_{ij\gamma'}}{\twoheadrightarrow} S_{ij\gamma'} \overset{[\subseteq,\subseteq]}{\longleftarrow} I' + O'$ *of Algo. 3 is a critical pair.*

穷尽性：任何形式为 $L_i + L\,j \twoheadrightarrow X \overset{[\subseteq,\subseteq]}{\longleftarrow} in(X) + out(X)$ 的临界对都可以由算法3生成。

正确性证明。余等化子是满态射，因此 $\varepsilon_{ij\gamma}$ 和 $\varepsilon_{ij\gamma'}$ 是满态射。它们的复合因此也是满态射。由于第 15 行的 if 语句，$I' \overset{\subseteq}{\to} S_{ij\gamma'} \overset{\subseteq}{\leftarrow} O'$ 是符合带接口的预临界对定义要求的 ma-余跨图。此外，匹配 $\iota_1; \varepsilon_{ij\gamma}; \varepsilon_{ij\gamma'}$和 $\iota_2; \varepsilon_{ij\gamma}; \varepsilon_{ij\gamma'}$都是单态射，因为粘合方案 γ和 γ′ 不会粘合属于同一超图的节点和超边。粘合方案 γ还会至少粘合一对超边（第 4 行）。因此，根据命题 3.6，所得的每个结果都是临界对。

穷尽性证明。设 $L_i + L\,_j \overset{\varepsilon}{\twoheadrightarrow} X \overset{[\subseteq,\subseteq]}{\longleftarrow} in(X) + out(X)$ 是一个带接口的预临界对。
超图的满态射在节点和超边上都是满射，因为超图范畴是预层范畴。

因此，X 的每个节点和每条超边在 ε 下都有非空原像集。此外，每个在 ε 下的原像集都满足：(i) 至多包含两个元素；(ii) 若包含两个元素，它们分别来自 L_1 和 L_2。这是因为 ι_i；ε 和 ι_j；ε 都是单态射。

我们构造一个超图 γ，其节点由大小为 2 的原像集 $\varepsilon^{-1}(v)$ 给出，其中 $v \in Nodes(X)$；其超边由大小为 2 的原像集 $\varepsilon^{-1}(e)$ 给出，其中 $e \in$ 超边(X)。它带有两个超图同态 $F_1 : \gamma \to L_1 + L_2$ 和 F_2：$\gamma \to L_1 + L_2$,，使得 F_1 将一个原像集映射到它在 L_1 中的元素，而 F_2 将一个原像集映射到它在 L_2 中的元素。我们得到一个粘合方案 (F_1,F_2)，而 X 是 (F_1, F_2) 的粘合。
由于该粘合方案会产生一个临界对，因此它满足第 3 节中命题的必要条件。具体而言：

1. 如果它粘合边，则这些边分别来自 L_1 和 L_2。2. 它至少分别从 L_1 和 L_2 粘合一对超边。
3. 如果它粘合节点，则这些节点要么是所粘合超边的源/目标，要么分别是 L_1 和 L_2 的输入/输出。

这些条件分别由算法3实现，具体对应第2行和第10行、第4行以及第10行。因此，我们可以得出结论：粘合方案 (F_1,F_2) 所规定的超边和节点的合并由算法3实现。

□

我们使用 Haskell[5], 实现算法3，并通过非交换双幺半群的示例 [5,第6节1]进行测试。虽然有22个临界对，但该实现输出了58个临界对。这是由同构粘合方案 γ,γ′造成的重复所致。目前，我们的实现不会检查超图是否同构。

算法 3：枚举所有临界对的算法

输入：重写规则 $\rho = \{L_i \xleftarrow{f_i} K_i \xrightarrow{g_i} R_i\}_{i\in I}$

输出：带有接口 $\{\{L_i + L^j$ 的满态射 $\xrightarrow{\varepsilon} S_{ij\gamma} \leftarrow I + O\}_{\gamma\in I_{ij}}\}_{(i,j)\in I^2}$

1 对于 $(i, j) \in I^2$ 执行

2 对于 $\gamma \in$ 导出超图($\bigcup_{l\in\Sigma}$ enumerateIndependentEdgeSets(

3 {e | e ∈超边(L_i),标签(e) $=l$},{e | e ∈超边(L j), 标签(e) $=l$})) do

4 if γ至少有一条超边 then

5 $(S_{ij\gamma},\varepsilon_{ij\gamma}) = \text{coeq}_\gamma(p_1^\gamma;\iota_1, p_2^\gamma;\iota_2)$;

/* 余等化子 (γ $\underset{p_2^\gamma;\iota_2}{\overset{p_1^\gamma;\iota_1}{\rightrightarrows}} L_i + L_j \xrightarrow{\varepsilon_{ij\gamma}} S_i j_\gamma$ 在 Hypσ) */

6 $I_1 = \text{in}(S_{ij\gamma}) \cap \varepsilon_{ij\gamma}(\iota_1(\text{in}(L_i)))$;

7 $I_2 = \text{in}(S_i j_\gamma) \cap \varepsilon_{ij\gamma}(\iota_2(\text{in}(L\ j)))$;

8 $O_1 = \text{out}(S_i j_\gamma) \cap \varepsilon_{ij\gamma}(\iota_1(\text{out}(L_i)))$;

9 $O_2 = \text{out}(S_i j_\gamma) \cap \varepsilon_{ij\gamma}(\iota_2(\text{out}(L\ j)))$;

10 对于 $\gamma' \in$ 导出超图(enumerateIndependentEdgeSets(I_1,O_2)+

11 enumerateIndependentEdgeSets(I_2,O_1)) 执行

12 $(S_{ij\gamma'},\varepsilon_{ij\gamma'}) =$ 余等化子$_{\gamma'}(p^{\gamma}{}_1{}', p^{\gamma}{}_2{}')$;

/* 余等化子 (γ' $\underset{p_2^{\gamma'}}{\overset{p_1^{\gamma'}}{\rightrightarrows}} S_i j_\gamma \xrightarrow{\varepsilon_{ij\gamma'}} S_i j_{\gamma'}$ 在 Hypσ) 中 */

13 $I' =$ 属于($S_{ij\gamma'}$);

14 $O' =$ 来自($S_{ij\gamma'}$);

15 如果 $I' \xrightarrow{\subseteq} S_{ij\gamma'} \xleftarrow{\subseteq} O'$ 是一个 ma-余跨图，那么

16 生成 $L_i + L_j \overset{\varepsilon_{ij\gamma};\ \varepsilon_{ij\gamma'}}{\twoheadrightarrow} S_i j_{\gamma'} \xleftarrow{[\subseteq,\subseteq]} I' + O'$;

17 **end**

18 **end**

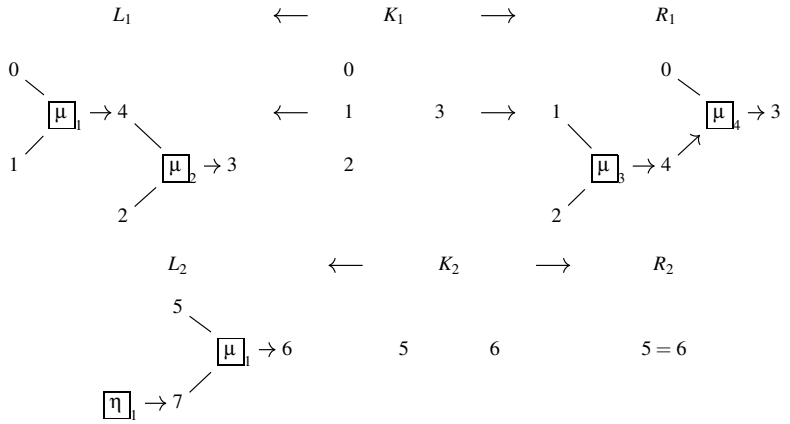
19 **end**

20 **end**

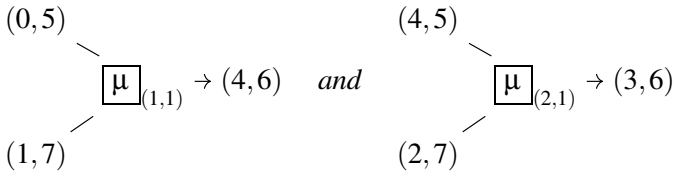
21 结束

例 3.10。我们计算以下一对规则所关联的临界对，这对规则取自非交换双幺半群的示例 *[5,*第 6 节。*1]*。

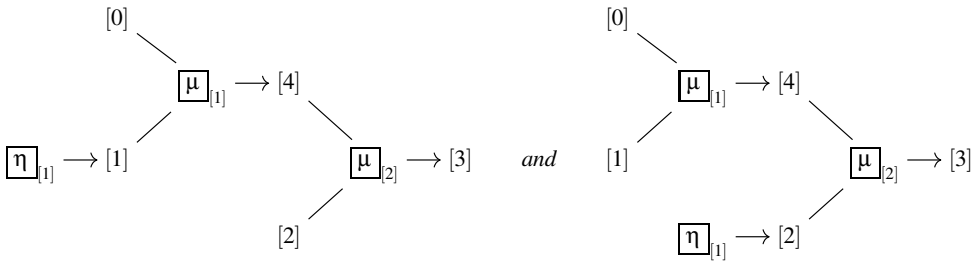
[5]在线获取：https://github.com/GuiSab/hypergraphrewriting



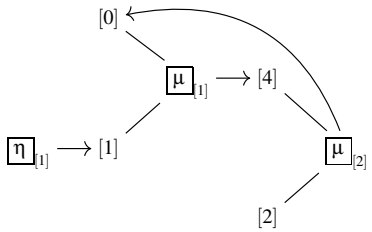
我们首先枚举与标签 μ相关联的独立边集：共有三个独立边集，分别为 $\{\}, \{(\mu_1, \underline{\mu}_1)\}$ 和 $\{(\mu_2, \underline{\mu}_1)\}$。标签η只有一个独立边集，即空集。因此，对于非空超边，我们有2种粘合方案：



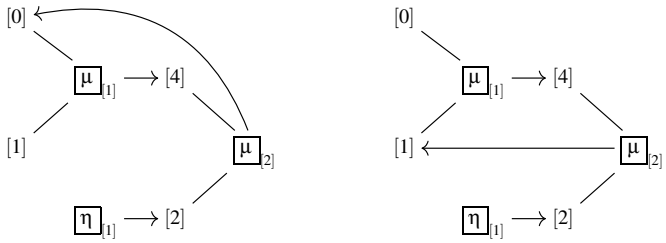
与粘合方案相关的粘合如下：



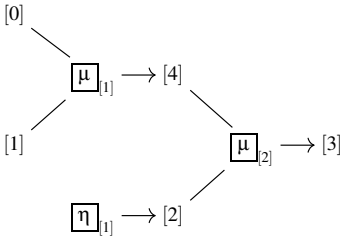
对于第一个粘合，我们计算 $I_1 = \{[0],[2]\}$、$I_2 = \{[0]\}$、$O_1 = \{[3]\}$、$O_2 = \{\}$。节点上的唯一独立边集是 $\{\}$ 和 $\{([0],[3])\}$。与 $\{([0],[3])\}$ 相关的粘合不是无环的：



因此，我们只得到由第一次粘合产生的临界对。对于第二次粘合，我们计算 $I_1 = \{[0],[1]\}$、$I_2 = \{\}$、$O_1 = \{[3]\}$、$O_2 = \{3\}$。节点上的独立边集为 $\{\}, \{([0],[3])\}$ 和 $\{([1],[3])\}$。与 $\{([0],[3])\}$ 和 $\{([1],[3])\}$ 相关的粘合不是无环的：



因此，我们只得到由第二次粘合产生的临界对：



4 优化

算法3 实现了两重粘合过程：首先粘合超边，其次粘合输入/输出。事实上，我们可以证明，出于临界对分析（以及局部合流性检查）的目的，第二步是多余的。

设 $S_{ij\gamma}$ 为算法3中超边的粘合，$S_{ij\gamma'}$ 为在 $S_{ij\gamma}$ 上对节点进行的粘合。

命题 4.1. 如果 $S_i j_{\gamma'}$ 产生一个临界对，那么 $S_{ij\gamma}$ 也会产生一个临界对。

证明. 如果 $S_{ij\gamma'}$ 是单配无环的，那么未粘合任何节点的超图 $S_{ij\gamma}$ 不可能有环；此外，它也满足单配条件。 □

现在假设 $S_{ij\gamma'}$ 产生了一个临界对（因此 $S_{ij\gamma}$ 也产生了一个临界对）。

命题 4.2。$I \to S_{ij\gamma} \leftarrow O$ 中的任意凸匹配都会在 $I' \to S_i j_{\gamma'} \leftarrow O'$ 中诱导出一个凸匹配。

证明。设 $m : L \to S_{ij\gamma}$ 是一个凸匹配。我们将证明 $m; \varepsilon_{ij\gamma'} : L \to S_{ij\gamma'}$ 也是一个凸匹配。

设 e_1 和 e_2 是 L 的两条超边，满足 $(m; \varepsilon_{ij\gamma'})(e_1) = (m; \varepsilon_{ij\gamma'})(e_2)$，也就是说，$\varepsilon_{ij\gamma'}(m(e_1)) = \varepsilon_{ij\gamma'}(m(e_2))$。必有 $m(e_1) = m(e_2)$，因为 $\varepsilon_{ij\gamma'}$ 只粘合节点，因此它在超边上是单态射。由此可得 $e_1 = e_2$，因为 m 是单态射。因此，$m; \varepsilon_{ij\gamma'}$ 在超边上是单态射。

设 v_1 和 v_2 是 L 的两个节点，使得 $(m; \varepsilon_{ij\gamma'})(v_1) = (m; \varepsilon_{ij\gamma'})(v_2)$，即 $\varepsilon_{ij\gamma'}(m(v_1)) = \varepsilon_{ij\gamma'}(m(v_2))$。

假设 $m(v_1) \neq m(v_2)$。$\varepsilon_{ij\gamma'}$ 会将 $m(v_1)$ 与 $m(v_2)$ 粘合。根据 $\varepsilon_i j_{\gamma'}$ 的构造，我们要么有 $m(v_1)$ 是 $S_i j_\gamma$ 的一个输入，且 $m(v_2)$ 是 $S_{ij\gamma}$ 的一个输出；要么有 $m(v_1)$ 是 $S_{ij\gamma}$ 的一个输出，且 $m(v_2)$ 是 $S_i j_\gamma$ 的一个输入。不妨设为第一种情况。由于 m 是凸匹配，在 m 的像中存在一条从 $m(v_1)$ 到 $m(v_2)$ 的路，由 $[m(e_1) \cdots、 m(e_n)]$ 组成。将 $\varepsilon_{ij\gamma'}$ 应用于该路后，我们得到一条从 $\varepsilon_i j_{\gamma'}(m(v_1))$ 到 $\varepsilon_i j_{\gamma'}(m(v_2)) = \varepsilon_i j_{\gamma'}(m(v_1))$ 的路，即 $[\varepsilon_{ij\gamma'}(m(e_1)) \cdots、 \varepsilon_i j_{\gamma'}(m(e_2))]$。$S_i j_{\gamma'}$ 将不是无环的，这与 $S_{ij\gamma'}$ 产生有效临界对的假设矛盾。因此，我们证明了 $m(v_1) = m(v_2)$。于是，由于 m 是单态射，我们可推出 $v_1 = v_2$。因此，$m; \varepsilon_{ij\gamma'}$ 在节点上是单态射。

$m; \varepsilon_{ij\gamma'}$ 是单态射，因此我们可以推出 $m; \varepsilon_{ij\gamma'} : L \to S_{ij\gamma'}$ 是一个凸匹配。 □

推论 4.3。 $\mathrm{I} \to \mathrm{S}_{ij\gamma} \leftarrow \mathrm{O}$ 上的任意重写序列都会诱导 $\mathrm{I}' \to \mathrm{S}_{ij\gamma'} \leftarrow \mathrm{O}$ 上的一个重写序列 $'$.

Cor推论 4.4。若 $\mathrm{I} \to \mathrm{S}_{ij\gamma} \leftarrow \mathrm{O}$ 产生一个可汇合的临界对，那么 $\mathrm{I}' \to \mathrm{S}_{ij\gamma'} \leftarrow \mathrm{O}'$ 也会产生一个可汇合的临界对 .

根据推论 4.4，只需枚举仅粘合超边的临界对，即可确定一个左连通重写系统是否具有局部合流性。算法 4 枚举了一个足够的临界对子集，用以判定局部合流性。

Algorithm 4: An algorithm for enumerating sufficient critical pairs

Input : rewrite rules $\rho = \{L_i \xleftarrow{f_i} K_i \xrightarrow{g_i} R_i\}_{i \in I}$

Output: epimorphisms with interface $\{\{L_i + L_j \overset{\varepsilon}{\twoheadrightarrow} S_{ij\gamma} \leftarrow I + O\}_{\gamma \in I_{ij}}\}_{(i,j) \in I^2}$

for $(i,j) \in I^2$ **do**
  for $\gamma \in$ `InducedHypergraphs(`$\prod_{l \in \Sigma}$ `enumerateIndependentEdgeSets(`
  $\{e \mid e \in Hyperedges(L_i), label(e) = l\}, \{e \mid e \in Hyperedges(L_j), label(e) = l\}$) **do**
    if γ **has at least a hyperedge then**
      $(S_{ij\gamma}, \varepsilon_{ij\gamma}) = \mathtt{coeq}_\gamma(p_1^\gamma; \iota_1, p_2^\gamma; \iota_2)$;
      `/* the coequaliser of (`$\gamma \overset{p_1^\gamma;\iota_1}{\underset{p_2^\gamma;\iota_2}{\rightrightarrows}} L_1 + L_2 \overset{\varepsilon_{ij\gamma}}{\cdots\!\rightarrow} S_{ij\gamma}$ `in` $\mathbf{Hyp}_\Sigma$`)` `*/`
      $I = in(S_{ij\gamma})$;
      $O = out(S_{ij\gamma})$;
      if $I \overset{\subseteq}{\Rightarrow} S_{ij\gamma} \overset{\subseteq}{\Leftarrow} O$ **is a ma-cospan then**
        yield $L_i + L_j \overset{\varepsilon_{ij\gamma}}{\twoheadrightarrow} S_{ij\gamma} \xleftarrow{[\subseteq,\subseteq]} I + O$;
      end
    end
  end
end

5 结论与未来工作

本文提出了一种算法，用于枚举给定左连通凸 DPOI 重写系统的所有临界对。我们证明了该算法的正确性和穷尽性。该算法围绕粘合这一双重过程展开：首先合并超边，其次合并输入/输出。我们还提出了对该算法的一种优化，仅合并超边。

我们关注所提算法（算法3与算法4）的复杂度分析，尤其是对优化有效性的评估。左连通性在我们的研究中至关重要，但或许可以借助形式路径扩展 [5]，将算法3扩展到非左连通重写系统。另一个未来方向是扩展到幺半闭范畴中的字符串图，对此DPOI 重写已有研究 [1]，但临界对分析尚未建立。

参考文献

[1] Mario Alvarez-Picallo、Dan R. Ghica、David Sprunger 和 Fabio Zanasi（2022）：幺半闭范畴的重写。载于 Amy P. Felty（编）：第七届形式结构与计算及演绎国际会议——

FSCD 2022，2022年8月2日至5日，以色列海法，LIPIcs 228，Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，第29:1–29:20页，doi:10.4230/LIPICS.FSCD.2022.29。[2] Franz Baader 和 Tobias Nipkow（1998）：项重写及相关内容。剑桥大学出版社，doi:10.1017/CBO9781139172752。[3] Filippo Bonchi、Fabio Gadducci、Aleks Kissinger、Pawel Sobocinski 和 Fabio Zanasi（2022）：字符串图重写理论 I：使用 Frobenius 结构进行重写。《ACM 杂志》69(2)，第14:1–14:58页，doi:10.1145/ 3502719。[4] Filippo Bonchi、Fabio Gadducci、Aleks Kissinger、Pawel Sobocinski 和 Fabio Zanasi（2022）：字符串图重写理论 II：使用对称幺半结构进行重写。《数学结构与计算机科学》32(4)，第511–541页，doi:10.1017/S0960129522000317。[5] Filippo Bonchi、Fabio Gadducci、Aleks Kissinger、Pawel Sobocinski 和 Fabio Zanasi（2022）：字符串图重写理论 III：有无 Frobenius 结构时的合流性。《数学结构与计算机科学》32(7)，第829–869页，doi:10.1017/S0960129522000123。[6] Kristopher Born 和 Gabriele Taentzer（2016）：合并图变换的临界对分析算法。载于 Rachid Echahed 和 Mark Minas（编）：《图变换》，施普林格国际出版，查姆，第118–134页，doi:10.1007/978-3-319-40530-8_8。[7] H. Ehrig、M. Pfender 和 H. J. Schneider（1973）：图文法：一种代数方法。载于：第14届开关与自动机理论研讨会（swat1973），第167–180页，doi:10.1109/SWAT.1973.11。[8] Hartmut Ehrig、Karsten Ehrig、Ulrike Prange 和 Gabriele Taentzer（2006）：代数图变换基础。施普林格，柏林，doi:10.1007/3-540-31188-2。OCLC：79651676。[9] Marcelo P. Fiore 和 Marco Devesas Campos（2013）：有向无环图的代数。载于 Bob Coecke、Luke Ong 和 Prakash Panangaden（编）：《计算、逻辑、博弈与量子基础：Samson Abramsky 的多重面向——献给 Samson Abramsky 六十岁生日的论文集》，《计算机科学讲义》7860，施普林格，第37–51页，doi:10.1007/ 978-3-642-38164-5_4。[10] Dan R. Ghica（2013）：延迟不敏感异步电路的图示推理。载于 Bob Coecke、Luke Ong 和 Prakash Panangaden（编）：《计算、逻辑、博弈与量子基础：Samson Abramsky 的多重面向——献给 Samson Abramsky 六十岁生日的论文集》，《计算机科学讲义》7860，施普林格，第52–68页，doi:10.1007/ 978-3-642-38164-5_5。[11] Ivaylo Hristakiev（2018）：图编程语言的合流性分析。博士论文，约克大学，英国。[12] Andr´e Joyal 和 Ross Street（1991）：张量演算的几何，I。《数学进展》88(1)，第55–112页，doi:10.1016/0001-8708(91)90003-P。[13] Yves Lafont（2003）：迈向布尔电路的代数理论。《纯粹与应用代数杂志》184(2)，第257–310页，doi:10.1016/S0022-4049(03)00069-0。[14] Samuel Mimram（2010）：计算二维重写系统中的临界对。载于 Christopher Lynch（编）：《第21届重写技术与应用国际会议论文集》，RTA 2010，2010年7月11日至13日，爱丁堡，苏格兰，英国，LIPIcs 6，Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，第227–242页，doi:10.4230/LIPICS.RTA.2010.227。[15] Grzegorz Rozenberg（1997）：《图文法与图变换计算手册》。世界科学出版社，doi:10.1142/3303。[16] P. Selinger（2010）：幺半范畴图形语言综述，第289–355页。施普林格柏林海德堡，柏林，海德堡，doi:10.1007/978-3-642-12821-9_4。