Agda 中抽象重写的形式化

Samuel Arkle ✉阿巴拉契亚州立大学，美国**Andrew Polonsky** ✉阿巴拉契亚州立大学，美国

摘要

我们在 Agda 证明助手中给出了抽象重写系统（抽象重写系统）的构造性形式化，重点关注项重写中的标准结果。我们定义了与终止性和合流性相关的概念分类，并研究这些概念与其经典对应概念之间的关系。我们识别并在可能之处消除标准抽象重写系统结果证明中对经典逻辑的使用。我们的分析对经典的终止性与合流性判据作出了改进和适度推广。我们研究了源于良基关系概念不同表述的若干终止性概念之间的逻辑关系。我们以 λ演算的形式化示例说明了我们所开发的抽象重写系统的普遍适用性。

2012年 ACM 主题分类 计算理论 → 类型论

关键词与短语 抽象重写、形式化、合流性、强正规化、纽曼引理、良基性

数字对象标识符 10.4230/LIPIcs...

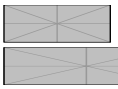
arXiv:2603.10936v1 [cs.LO] 11 Mar 2026

1 引言

我们介绍了一个 Agda 形式化成果，涵盖 Bezem、Klop 和 De Vrijer 在其著作中阐述的抽象重写系统（ARS）基础理论；下文将该著作简称为 Terese [8]。这项工作是阿巴拉契亚州立大学开发形式化编程语言理论（PL）库这一更大项目的一部分。项重写系统（TRS）在编程语言中发挥着基础性作用，而抽象重写系统涵盖了关于一般重写关系的基本事实，因此，确立这些事实是推进这一更广泛项目所需的必要基础设施之一。因此，本项工作可视为在“启动”阿巴拉契亚州立大学新的形式化 PL 仓库。 [6]

这一更广泛的目标也决定了我们方法中的主要设计选择。重写理论关注对计算进行细致入微的分析，因此，形式化它最自然的工具是基于 Curry–Howard同构的类型论，而 Agda 实现了这种类型论。使用这种语言编写的证明会自动具有可计算性，并隐含地携带实现从假设到论断结论之变换的代码。例如，经过类型检查、证明给定项重写系统具有 Church–Rosser 性质的证明，会自动生成一个函数，用于计算任意两个可转换项之间的公共约化。因此，Agda 中的证明不仅仅是对正确性的验证，也是对构造性证明中隐含的概念性算法的实现。能够按需执行这类计算，有助于发展复杂的本体和理论。这使 Agda 成为我们更大项目特别有吸引力的工具。

要坚持这一范式，证明就必须是构造性的。此外，我们还必须避免使用任何公理或假设，例如函数外延性、恒等证明唯一性、单值性等。尽管抽象重写系统的大多数结果确实都是构造性的，但标准的论述（包括 [8], ）却大量使用经典逻辑。因此，我们的论文可以被视为对初等抽象重写系统理论进行的全面构造性发展。



© Samuel Arkle 和 Andrew Polonsky；依据知识共享许可协议 CC-BY 4.0 授权。莱布尼茨国际信息学论文集，Dagstuhl 城堡 – 莱布尼茨信息学中心，Dagstuhl 出版社，德国

XX:2 在 Agda 中对抽象重写进行形式化

1.1 动机

在继续之前，我们先来考虑两种形式化场景，在这些场景中，构造性的抽象重写系统库会很有用。

▶ **例 1**（带类型的 λ 演算的形式化）。假设有人正在对带类型的 λ 演算的标准元理论进行形式化。为了证明合流性，可以先为无类型 λ 演算确立 Church–Rosser 定理（合流性），然后通过主体约化定理继续证明。然而，假设强正规化已经经过独立验证。利用纽曼引理，可以从弱菱形性质推出合流性——而弱菱形性质通常比完整的 Church–Rosser 定理更容易证明。

▶ **例 2**（类型的商）。假设我们希望构造一个一阶签名 S上的代数理论 T 的初始模型。（例如，读者可以设想有限生成元集合上的自由交换半环。）经典定义涉及项代数 S^* 关于由 T导出的等价关系 $\approx_T$ 所取的商。在纯类型论中直接编码商并不容易。尽管存在若干解决方案，例如高阶归纳类型 [1], 、商归纳-归纳类型 [4],、立方类型论 [12], 等，但它们都需要通过引入新的语言构造来扩展类型论，从而增加了复杂性方面的额外成本。

现在，假设 T 具有一个合流表示。也就是说，假设 S^* 上的等价关系与由一个合流且终止的重写系统 $\to_R$: $S^* \to S^* \to$ **集合**生成的转换相一致。这样，就可以将 $S^*/\approx_T$ 替换为 R- 范式的类型。（在半环示例中，这些范式将是按照单项式上的适当顺序排列的多项式表达式。）这可以定义为一个纯依赖类型

$$NF(R) \quad := \quad (\Sigma n : S^*)(\Pi x : S^*)\neg(n \to_R x)$$

该类型以如下意义编码了商：函数 $S^*/\approx_T \to A$ 的类型与从 $NF(R)$ 到 A 的普通函数类型同构。实际上，商类型的消去规则要求 T 中的每个等式都得到验证，才能定义函数 f: $S^*/\approx_T \to A$。任何这样的函数都可以平凡地特化为 $NF(R)$。反过来，从$NF(R)$ 到 A 的普通函数可以通过对规范化归约序列进行归纳，扩展到 S^* 的全部元素。合流性确保 T-相等的元素始终映射到相同的值。

因此， $S^*/\approx_T$与 $NF(R)$同构。
由此还可推论，初始模型中的相等关系是可判定的。

这些例子表明，对基本抽象重写系统结果进行完全可计算的实现，有助于解决编程语言理论中的许多问题。

1.2 形式化原则

Agda 既是一种编程语言，也是一种交互式定理证明器。作为编程语言，它可以被视为扩展了依赖类型的 Haskell，这极大地拓展了类型系统的表达能力，使复杂的规格能够直接嵌入所编写函数的类型中。作为定理证明器，Agda 是 Martin-Löf 直觉主义类型论的一种实现，并严格遵循证明即程序范式。与这一类中的其他证明助手一样，它的逻辑基于将公式编码为扩展型 λ 演算中的类型，而底层的项则充当这些公式的证明。Agda 具有类型化 λ 演算通常具备的元理论性质，包括主题归约、强正规化和正则性。

从字面意义上说，Agda 中的证明就是依赖类型函数式程序。对递归类型或推导树进行结构归纳所得的证明，会表示为通过模式匹配定义的递归函数，这对函数式程序员而言十分熟悉。依赖类型系统支持一种独特的交互式编程风格：编写代码往往归结为在给定上下文中构造特定类型的值。只要代码不使用经典逻辑（即排中律）或其他公理（例如函数外延性），这些作为程序的证明就能保证会进行计算。

为了明确地将基本抽象重写的结果以规范的方式形式化，使其能够用于构建更大型的形式化编程语言理论库，我们在开发过程中遵循了以下原则。

- 不使用函数外延性；不使用单值性、恒等证明唯一性、K 公理或任何其他与相等有关的
- 假设；尽量减少对经典逻辑的使用，并在无法避免经典逻辑的每处明确列出可判定性假设；忠实遵循 Curry–Howard 范式，使我们的证明在每个具体实例中都能计算出明确
- 的见证。

1.3 可判定性

在若干处，为了使证明得以成立，我们必须假设某些可判定性假设。这些假设始终都被明确标出，并且我们尽力将其数量减至最少。

回想一下，一个关系是强正规化的，当且仅当它的逆关系是良基的。 [8, Def.1.1.13]。由于正规化是抽象重写系统理论中的核心概念，我们特别关注良基性这一概念。标准的构造性定义甚至无法证明强正规化蕴含弱正规化（即每个对象都能通过至少一次有限归约到达一个范式），我们将在第 5 节末尾讨论并在 `ARS/Examples.agda` 中形式化这一点。因此，我们考察了构造性语境下这一概念的多种变体；它们的经典对应概念在逻辑上都等价。

我们还找到了若干类抽象重写系统，在这些系统中所需的经典原理自然成立。特别是，对于有限分支关系，SN 蕴含 WN 无需任何额外条件，只要求关系本身是可判定的：$Rxy \vee \neg Rxy$。例如，由含有限规则集的一阶项重写系统所诱导的抽象重写系统，既是有限分支的，也是可判定的。

我们工作的主要结论是，大多数关于抽象重写系统的主要结果都可以完全做到可计算，至少对于一阶项重写系统和 λ 演算中遇到的实际例子是如此。对于更为特殊的重写系统——例如 $\lambda\bot$ （它是合理 λ 理论的基础），或余归纳重写——这些可判定性假设不再成立，抽象重写系统框架的实用性也会相应降低。

1.4 贡献与相关工作

目前已有若干关于初等抽象重写系统理论概念的形式化工作，包括 Isabelle [14], 、PVS [13], 和 Rocq [10] 中的项目，以及 Agda 标准库 [3]。前三者都是相当成熟的大型项目，旨在为一般重写系统的终止性提供证明。我们的工作规模与 Agda 库中的重写开发更为接近，旨在作为一个新的形式化编程语言理论库的起点。除上述引用的工作外，我们的具体贡献如下。

- 对 Terese [8]，所述的初等抽象重写系统理论进行了原创性形式化，其中包含关系与逻辑的相关背景知识；构建了终止性与合流性属性的本体，并详细分析了它们之间的逻辑
- 关系；定义了新的抽象重写系统概念，以深化我们对这些关系的理解；对经典的合流性与终止性判据作了少量改进；考察了构造性语境下若干种不同的良基性概念。
-
-
-

1.5 论文结构

论文的结构大致对应于 Agda 代码的推进过程。下一节将阐述抽象重写系统理论所研究的关系的主要性质，为后续内容奠定基础。第 3 节概述我们对 [8] 中抽象重写系统章节的形式化。在此，我们还将讨论这一工作如何促使我们提出新的抽象重写系统性质，帮助我们克服过程中遇到的一些障碍。这些性质自然地契合重写范式。因此，第4节将着重进一步研究它们。由此，我们得到了更广泛的抽象重写系统概念体系，揭示了完备性条件（即同时具有强正规化性和 Church-Rosser 性）之间的相互关系。第 5 节聚焦于关系良基的含义，并辨明了在不同表述之间转换所需的经典原理。第 6 节以无类型 λ 演算为例，展示该库如何应用于更大的项目。最后，我们将简要总结并提出未来工作的建议。

本文中的已知结果均未给出证明，仅注明相应的 Agda 代码。 [6] 对于新的初等命题，通常会附上一句非正式的说明。当我们的研究过程中发现新的想法或联系时，通常会在专门的注释或正文中加以讨论。这些选择体现了我们贡献的性质，以及为更大型的项目奠定基础的目标。

2 定义

为确保讨论清晰准确，下面给出相关定义。每个小节都会指出包含相应定义形式化内容的 Agda 文件。

▶ **定义 3。一个抽象重写系统（ARS）是一个结构** $A = (A, R_\alpha)$，**其中** A**是一个集合，** R_α **是** A**上的二元关系集合。**

在本文余下部分中，设 A 为一个固定集合， R 为 A上的一个关系。对于 a， $b \in A$，$a \equiv b$ 表示严格相等，在 Agda 中用通常的恒等类型编码。

2.1 闭包算子(Relations/ClosureOperators.agda)

▶**记号 1.** 以下记号是标准记号。
1. R^r **表示** R**的自反闭包。2.** R^s **表示** R**的对称闭包。3.** R^+ **表示** R**的传递闭包。4.** $R^* = (R^r)^+$ **表示** R**的自反传递闭包。5.** $R^= = (R^s)^*$ **表示由** R**生成的等价关系。**

在抽象重写的语境中，我们通常将 $a \longrightarrow_R b$ 写作 $(a, b) \in R$。我们还将 $a \longrightarrow_R^* b$ 写作 $(a, b) \in R^*$，将 $a \longleftarrow_R b$ 写作 $(b, a) \in R$，并将 $a \not\longrightarrow_R b$ 写作 $(a, b) \notin R$。当关系 R 可由上下文明确确定时，我们可以省略相应的下标。

2.2 正规化与合流性（ARS/Properties.agda）

我们将合流性和正规化等重写理论概念定义为关于 A 的谓词，用于说明给定元素 $a \in A$ 在局部是否满足某项性质。随后对 a 进行全称量化，即可得到关系 R 的相应全局性质

▶ **定义 4.** 设 $a \in A$。

- $a \in \mathrm{NF}_R$：a 是**范式**，如果 $a \not\longrightarrow_R b$ 对任意 $b \in A$ 。
- $a \in$ **弱正规化**$_R$：a 是**弱正规化**，如果 $a \longrightarrow_R^* b$ 对某个 $b \in \mathrm{NF}_R$。
- $a \in$ **强正规化**$_R$：a 是**强正规化**，如果 a 关于以下关系的逆是可达的：R，参见第 *5* 节。也就是说，$a \in$ 强正规化$_R$ 如果对每个 $b \in A$，$a \longrightarrow_R b$ 蕴含 $b \in$ 强正规化$_R$。这是一个归纳定义，其基础情形取自那些 a 满足 $a \in \mathrm{NF}_R$。

▶ **定义 5**。设 $a \in A$。

- $a \in$ **弱合流性**$_R$：a 是 **弱丘奇-罗瑟 (**或 弱合流**)**，如果对所有 b，$c \in A$ ，

$$c \longleftarrow_R a \longrightarrow_R b \implies \exists d.\, c \longrightarrow_R^* d \longleftarrow_R^* b$$

- $a \in$ **合流性**$_R$：a 是 **Church-Rosser (**或 合流**)**，如果对所有 b，$c \in A$，

$$c \longleftarrow_R^* a \longrightarrow_R^* b \implies \exists d.\, c \longrightarrow_R^* d \longleftarrow_R^* b$$

- $a \in$ **合流性**$^{\leq 1}{}_R$：a 是 **次交换** ，如果对所有 b，$c \in A$，

$$c \longleftarrow_R^* a \longrightarrow_R^* b \implies \exists d.\, c \longrightarrow_R^r d \longleftarrow_R^r b$$

- $a \in$ **正规形性质**$_R^=$：a 具有 **转换正规形性质** ，如果对所有 $b \in \mathrm{NF}_R$，

$$a \longrightarrow_R^= b \implies a \longrightarrow_R^* b$$

- $a \in$ **正规形性质**$_R^\rightarrow$：a 具有 **归约正规形性质** ，如果对所有 $b \in \mathrm{NF}_R$，

$$c \longleftarrow_R^* a \longrightarrow_R^* b \implies c \longrightarrow_R^* b$$

- $a \in \mathrm{UN}_R^=$：a 具有**转换唯一正规形性质**，当且仅当对于 a，$b \in \mathrm{NF}_R$，

$$a \longrightarrow_R^= b \implies a \equiv b$$

- $a \in \mathrm{UN}_R^\rightarrow$：$a$ 具有**归约唯一正规形性质**，当且仅当对于所有 b，$c \in \mathrm{NF}_R$，

$$c \longleftarrow_R^* a \longrightarrow_R^* b \implies b \equiv c$$

▶ **记号 2**。对于上文定义的每个谓词 X_R，如果 $\forall a.\, a \in \mathrm{X}_R$，我们记作 $\mathbf{X}(R)$。当 R 由上下文蕴含时，我们将 $\mathbf{NP}^\rightarrow$ 记作 $\mathbf{NP}^\rightarrow(R)$，其他情况亦同。

也就是说，对于 $\mathrm{X} \in \{\mathrm{WCR}、\mathrm{CR}、\mathrm{NP}^\rightarrow、\mathrm{UN}^=、\ldots\ldots\}$，$\mathbf{X}$ 表示命题 $\forall a. a \in \mathrm{X}_R$。

在 Terese [8], 中，NF 用于表示正规形性质 $\mathrm{NP}^=$，而我们选择使用 NF 来表示范式的集合。

▶**命题 6**。

1. 正规形性质$^=$ $\iff$ 正规形性质$^\rightarrow$
2. 唯一范式性$^=$ $\implies$ 唯一范式性$^\rightarrow$

本文中的所有证明都已在 Agda 中完全形式化，其中许多证明无需进一步阐述。全文中，我们通过给出 Agda 文件的名称，以及对应于各项论断的函数名称和类型，来标明代码库中的相关位置。

上述命题的证明形式化于 `ARS/Implications.agda`，如下所示。1. `NP`$_=$`↔NP : R isNP`$_=$ `↔ R isNP`2. `UN→UN`$^{\to}$`: R isUN → R isUN`$^{\to}$

▶ 注 7. UN$^{\to}$ $\implies$ UN$^{=}$ 这一蕴含关系并不成立，这一点可以从反例中看出 r-图 1 中的例 5。

2.3 序列与递归（`Relations/Seq.agda`, `ARS/Properties.agda`）

▶定义 8。

- 序列 是一个从自然数到 A的函数 s 。
- 若序列 s: $\mathbb{N} \to A$ 中的每个元素都与其前一项满足 R-关系，则称该序列是 **R-递减的**：$\forall k.s(k+1) \longrightarrow_R s(k)$。
- 若序列 $s : \mathbb{N} \to A$ 中的每个元素都与其后一项满足 R-关系，则称该序列是 **R-递增的**：$\forall k.s(k) \longrightarrow_R s(k+1)$。

▶定义9。

- $\underline{x \in CP_R}$: x 具有 **共尾性**，如果存在一个 R^r-递增序列$s : \mathbb{N} \to A$ 从 x 出发，使得 x 的每个归约式都归约到该序列中的某个元素。

$$x \longrightarrow_R^* y \implies \exists i.y \longrightarrow_R^* s(i)$$

注意上面的定义使用了自反闭包 R^r 。这使得归约序列 $s(i)$ 从某一点开始可以为空，这对于使该表述等价于 Terese 中的表述是必要的。（特别是，正规化 x 具有共尾性。）

▶定义10。

- $a \in A$ 是一个 s-**上界**，如果对所有 $i \in \mathbb{N}$， $s(i) \longrightarrow_R^* a$。
- s： $\mathbb{N} \to A$ 是**有界的**，如果存在一个 $a \in A$，它是一个 s-上界。

▶定义11。 以下是给定关系 R的全局性质。

- $\underline{\mathrm{BP}(R)}$: R 具有**有界性**，如果每个 R-递增序列都有界。
- $\underline{\mathrm{RP}(R)}$: R 具有**递归性**，如果对每个有界递增序列 s，都存在一个 $i \in \mathbb{N}$ 使得 $s(i) \longrightarrow_R^* x \implies x \longrightarrow_R^* s(i)$ 对所有 $x \in A$成立。
- $\underline{\mathrm{RP}^-(R)}$: R 具有**弱递归性**，如果对每个有界递增序列 s，其上界为 b，都存在一个 $i \in \mathbb{N}$ 使得 $b \longrightarrow_R^* s(i)$。

性质 **BP** 等价于 Terese [8] 中称为 **Ind** 的性质。更准确地说，**Ind** 还断言有限序列也存在一个上界，但这是显然的，因为有限归约序列的最后一项就是该序列的一个上界。

Terese [8] 还定义了以下性质，该性质未包含在我们的形式化中。

▶定义12。

- $\underline{\mathrm{Inc}(R)}$: R 在 存在一个“大小函数”时递增 $|-|$: $A \to \mathbb{N}$ 满足对所有a， $b \in A$， $(a \longrightarrow_R b) \implies |a| < |b|$。

要求存在一个将所有 A 映射到自然数的大小函数，在计算上可能代价很高。此外，这也使我们脱离了纯关系理论。上文定义的较弱性质 **菱形性质** 将在所有需要 **Inc** 的地方取而代之。更多讨论请参见第3.1小节。

▶**命题 13**。
1.Inc $\implies$ 菱形性质2.
菱形性质 $\iff$ 菱形性质$^{-}$

证明。
1. Inc意味着不存在有界的 R-递增序列，因此RP 自然成立。2. 参见
`RP-↔RP : R isRP- ↔ R isRP`中的 `ARS/Implications.agda`。◀

2.4 有限分支（`Relations/FinitelyBranching.agda`）

▶**定义 14**。
- $a \in \mathrm{FB}$: a 是**有限分支**，如果存在有限多个 $b \in A$ 使得 $a \longrightarrow_R b$。

我们对上述定义的 Agda 表述使用了 List 类型构造器，该构造器从 Agda 标准库 [2] 导入。

```
FB a = Σ[ xs ∈ List A ] (∀ b → R a b → b ∈List xs)
```

3 Terese [8]的形式化

本节是我们对 Terese [8] 进行形式化的指南，并解释我们相对原文所做的偏离。

3.1 定义的形式化（`Base.agda`）

如上一节所述，文件 `Properties.agda` 定义了大多数标准的终止性与合流性性质。文件 `Base.agda` 引入了额外的定义，这些定义是对 Terese [8] 第1章中的命题进行形式化所必需的。

以下是第2节中提到的我们所采用的定义之间差异的总结。

- SN：我们通过可达性的概念来定义它，而不是通过不存在无限归约序列来定义。两个定义都将正规化表述为逆关系的良基性。不过，这两个定义使用了不同的良基性概念。相关差异将在第 5 节中重点说明。NP$^{=}$ ：我们采用等价的 NP$^{\rightarrow}$ 定义。Inc：我们采用更一般的菱形性质。
-
-

如前所述，Inc 性质可能会提出计算开销较大的要求。然而，Inc 仅用于定理1.2.3，而证明所需的 Inc 的唯一关键方面是：不存在具有上界的 R-递增序列。我们证明，较弱的菱形性质（或其等价形式 RP^{-}）足以在该定理中替代 Inc（见第3.2小节）。这促使我们更仔细地考察菱形性质中出现的关于 $s(i)$ 的条件。该条件推广了范式的概念，我们将在第4节中对其进行详细探讨。

3.2 定理的形式化 `Theorems.agda`

▶ **定理 15.** *(纽曼引理)* **强正规化** $\wedge$**弱合流性** $\implies$ **合流性**

证明。 Terese 中给出了纽曼引理的三种证明 [8]。其中第一种和第三种证明使用了经典推理。第二种证明适合进行构造性形式化，如下列函数所示 `ARS-NewmansLemma.agda`：

```
NewmansLemma : R isSN → R isWCR → R isCR.
```

◀

我们在第4.2小节中给出纽曼引理的一个推广，其条件提炼自**菱形性质**。其余定理的形式化均见于 Theorems.agda。

▶**定理 16** (Terese [8]中的定理 1.2.2。**i)合流性** $\Longrightarrow$ **正规形性质**$^{=}$ $\Longrightarrow$ **唯一范式性**$^{=}$**ii)弱正规化** $\wedge$**唯一范式性**$^{=}$ $\Longrightarrow$ **合流性iii)合流性**$^{\leq 1}$ $\Longrightarrow$ **合流性**

证明。 参见 module Theorem-1-2-2见 Theorems.agda,

```
CR→NP : R isCR → R isNP_NP→UN : R isNP₌ → R isUN
ii     : R isWN × R isUN → R isCR
ii+    : R isWN × R isUN⁻→ → R isCR
iii    : subcommutative R → R isCR
```

◀

▶ 注 17. 在 $\mathtt{UN}^{\rightarrow} \wedge \mathtt{WN} \rightarrow \mathtt{CR}$ 中，使用唯一范式性$^{\rightarrow}$给出了（ii）的一个更一般的证明。

▶**定理 18** (Terese [8]中的定理 1.2.3。**i) 弱正规化** $\wedge$ **唯一范式性**$^{=}$ $\Longrightarrow$ **BPii) BP** $\wedge$ **Inc** $\Longrightarrow$ **强正规化iii) 弱合流性** $\wedge$**弱正规化** $\wedge$**Inc** $\Longrightarrow$ **强正规化iv) 合流性** $\Longleftrightarrow$ **CP** (只有在抽象重写系统是可数时，正向蕴含才成立。)

证明。 参见 module Theorem-1-2-3中 Theorems.agda,

```
i     : R isWN → R isUN → R isBP
i→    : R isWN → R isUN⁻→ → R isBP
iiSeq : R isWN → R isUN → R isRP → isWFseq- (~R R)
iii   : R isWN → R isWCR → R isRP- → dec SN → R isSN
iv    : R isCP → R isCR
```

◀

▶注 19。
(i) 在 $\mathtt{i}^{\rightarrow}$ 中，我们再次通过使用 **唯一范式性**$^{\rightarrow}$ 而不是 **唯一范式性**$^{=}$，(ii) 我们将**BP** 替换为更强的性质 **弱正规化** 和 **唯一范式性**$^{=}$，并将 **Inc** 替换为较弱的性质 **菱形性质**。对于原有性质 **BP** 和 **菱形性质**，我们可以推出相关性质 **半合流性** (**半合流性** 在第4节中定义，并进一步阐述了它与 **强正规化** 之间的关系)。关于 isWFseq$-$ 的定义和进一步讨论见第 5 节。

(iii) 在 iii中，我们将 **Inc** 替换为更一般的性质 **菱形性质**$^{-}$。我们还假设 **强正规化**是可判定的，因为这样便可以构造一个见证，用以判断某个元素是否具有 强正规化这一性质，而这对本证明至关重要。

4 层级

本节中的工作已在 Properties.agda 和 Implications.agda中形式化。本节考察抽象重写系统属性的层级。在编程语言理论中，**强正规化** 和 **合流性** 是两项基本性质，二者共同构成完备性的基础。因此，我们的分析围绕两大类主要性质展开：与终止性相关的性质，以及与合流性相关的性质。

如第3.1小节所述，我们提出以菱形性质作为 Inc 的替代方案。通过考察菱形性质，我们得出了以下“最小”性质族。这些性质有助于丰富我们的 ARS 层次。

▶ **定义 20.** 设 $a \in A$。

- $a \in \mathrm{MF}_R$: a 是一个**极小形式**，当且仅当 $a \longrightarrow_R^* b \implies b \longrightarrow_R^* a$ 对任意 $b \in A$。
- $a \in \mathrm{WM}_R$: a 是 **弱极小化**，当且仅当 $a \longrightarrow_R^* b$ 存在某个 $b \in \mathrm{MF}_R$。
- $a \in \mathrm{SM}_R$: a 是 **强极小化**，当且仅当以下情况之一成立： $a \in MF_R$，或每个与之相距一步的元素R a 都是强极小化。（此定义应按归纳方式理解。）
- $a \in \mathrm{SMseq}_R$: a 是 **序列强极小化**，当对每个 R-递增序列 s 从 a开始时，都存在一个 $i \in \mathbb{N}$ 使得 $s(i) \in \mathrm{MF}$。
- $a \in \mathrm{MP}_R$: a 具有 **极小形式性质**，当且仅当 $c \longleftarrow_R^* a \longrightarrow_R^* b \implies c \longrightarrow_R^* b$ 对于任意 $b \in \mathrm{MF}$。

▶ **命题 21**。 $\forall x.\, x \in \mathrm{NF} \implies x \in \mathrm{MF}$

证明。 参见， `NF ⊆ MF : ∀x → NFx → MFx`于 `Implications.agda`。 ◀

SMseq 仅在经典逻辑下与 SM 等价。以下命题将这些新定义与我们先前对菱形性质的定义联系起来。

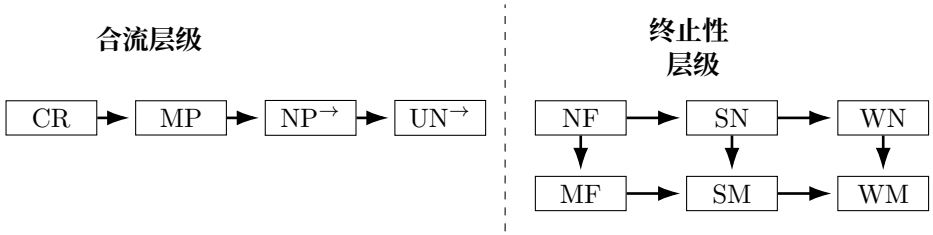
▶ **命题 22. SMseq $\iff$ 菱形性质 $\wedge$BP**

证明。 参见 `Implications.agda`,

```
RP∧BP→SMseq    : R isRP → R isBP → ∀ {x : A} → SMseq R x
RisSMseq→RisRP : (∀ {x : A} → SMseq R x) → R isRP
RisSMseq→RisBP : (∀ {x : A} → SMseq R x) → R isBP
```

◀

命题 21 表明，我们应将 MF 视为一种独立的终止性质。这促使我们探讨以下与完备性相关的、更广泛的概念分类。



该图中的蕴含关系已在 `Implications.agda` 中形式化。合流层级不需要经典假设，但终止性质层级则不然。由 SN 推出 WN 的蕴含关系依赖于性质「R-极小」的经典可判定性，因此其形式化形式为

```
SNdec→WN : (~R R) isMinDec → SN ⊆ WN
```

第 4.4 小节和第 5 节讨论了为何这一蕴含关系需要假设一个经典公理。证明 SM 蕴含 WM 时也需要同一性质。

▶**命题 23**。

1. $\forall x.\, x \in$ 最简形 $\cap$ 弱正规化 $\iff x \in$ 正规形
2. $\forall x.\, x \in$ 最简形 $\cap$ 强正规化 $\iff x \in$ 正规形

证明。 参见模块 `Normalizing-Implications`位于 `Implications.agda`,

```
MF∧WN↔NF : ∀ {x} → MF x × WN x ↔ NF x
MF∧SN↔NF : ∀ {x} → MF x × SN x ↔ NF x
```

◀

与上述命题相反，SN $\subset$ SM $\cap$ WN 的包含关系是严格的：在反例 4 中， $a \in$ SM $\cap$ WN，但 $a \notin$ SN。因此，在下表中，我们为 SM 与 WN 的组合单独设置一列。

4.1 完备性条件

现在我们来考察哪些性质组合足以逆转这些蕴含关系并实现完备性。下表展示了对于每种性质组合，是否能够得到合流性（左框）和强正规化（右框）。如果无法推出这些性质，我们会提供反例来说明还需要附加条件；这些反例标记为 CE-X。

第一张表探讨这些性质在何处局部成立，第二张表探讨这些性质在何处全局成立。弱合流性对第一张表中的反例成立，**弱合流性** 对第二张表中的反例成立**弱合流性** （提醒一下，粗体文本用于表示某性质全局成立）。

	WM		WN		SM		半合流性 ∧ 弱正规化		SN	
$UN^{\rightarrow}$	CE-4	CE-9	CE-4	CE-6	CR*	CE-8	CR*	CE-4	CR*	SN
$NP^{\rightarrow}$	CE-3	CE-8	CR	CE-6	CR*	CE-8	CR	SN	CR†	SN
MP	CR	CE-8	CR	CE-6	CR†	CE-8	CR	SN	CR†	SN
CR	CR	CE-8	CR	CE-6	CR	CE-8	CR	SN	CR	SN

表1 局部蕴含

	WM		**WN**		**SM**		**半合流性 ∧ 弱正规化**		**SN**	
$UN^{\rightarrow}$	CE-2 CE-8 **合流性**		CE-6 **合流性***		CE-8 **合流性**		**SN**		**合流性†**	**强正规化**
$NP^{\rightarrow}$	CE-3 CE-8		**CR**	CE-6	**CR***	CE-8	**CR**	**SN**	**合流性†**	**强正规化**
MP	**CR**	CE-8 **合流性**		CE-6 **合流性†**		CE-8 **合流性**		**SN**	**合流性† 规化**	**强正**
CR	**CR**	CE-8 **合流性**		CE-7	**CR**	CE-8 **合流性**		**SN**	**CR**	**SN**

表 2 全局蕴含。

* 此蕴含还需要**弱合流性**。

† 这要求具备**WCR**，或者满足经典蕴含 SN → WN，或满足 SM → WM。

该表应作如下解读：例如，若某个元素具有性质 $NP^{\rightarrow}$，那么根据该表，若要获得合流性 CR，至少还需要具备性质 WN。反例3表明，WM 与 $NP^{\rightarrow}$ 的组合不足以保证 CR。

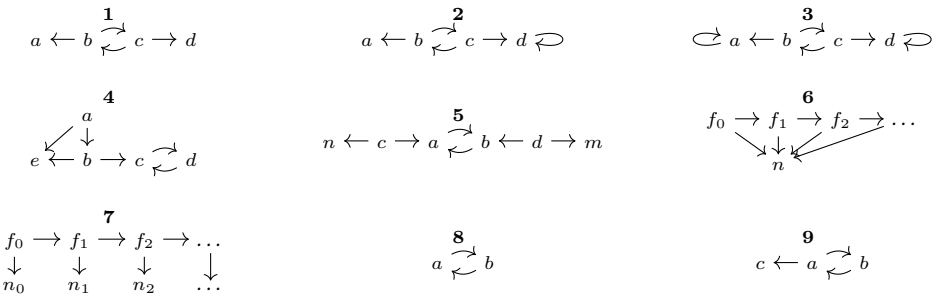


图 1反例

▶注24。 以下是蕴含表中的关键要点。1. 性质 **WCR** 有时可以替代经典性质，而该经典性质是SN $\Longrightarrow$ WN （类似地，SM $\Longrightarrow$ WM）。

2. WN ∧ UN$^{\rightarrow}$ $\Longrightarrow$ CR 的蕴含关系不需要任何经典假设或 WCR，这一点不同于 SN ∧ UN$^{\rightarrow}$。3. WN ∧UN$^{\rightarrow}$ $\Longrightarrow$ CR 仅在这些性质是全局性质时成立。反例4表明，WCR ∧ WN ∧UN$^{\rightarrow}$ $\not\Rightarrow$ CR。4. SN 需要 SM 和 WN，见第4.3小节。5. 我们将纽曼引理推广为 SM ∧ WCR $\Longrightarrow$ CR。

4.2 广义纽曼引理

研究正规化分类法的一个有趣结果，是对纽曼引理的推广。

▶ **命题25。弱合流性** ∧半合流性 $\Longrightarrow$ 合流性

证明。 见： `LocalNewmansLemmaRecurrent : R isWCR → SM ⊆ CR` ◀

这个证明沿用了 Terese [8]，中纽曼引理的第二个证明，但将 SN 替换为较弱的假设 SM。为展示其优势，考虑具有以下规则的 TRS：$p(a) \to p(b)$、 $p(b) \to p(a)$、 $f(p(a)\ p(a)) \to k$，以及 $f(p(b)\ p(b)) \to k$。由于 $p(a) \leftrightarrow p(b)$ 会产生无限归约，**SN** 不成立。尽管如此，该系统满足**WCR** 和**SM**，因此即使没有**SN**，该命题仍能推出**CR** 。

4.3 SM、WN 与 SN 之间的关系

从表格中可以得到一个重要结论：当具备 SM、WN 和 NP$^{\rightarrow}$ 这些性质时，我们可以得到强终止性和合流性。具备**WN** 和**SM**时，我们也可以得到完备性。为了说明**SM** ∧ **WN** $\Longrightarrow$ **SN** ，我们以证明 SM ∧ WN ∧ NP$^{\rightarrow}$ $\Longrightarrow$ SN 为基础。如果**WN** 成立，那么我们的证明就不再需要性质 NP$^{\rightarrow}$。这一系列证明可见于以下函数：

```
WN∧NP∧SM→SN : ∀ {x} → WN x → NP x → SM x → SN x
isWN∧SM→SN : R isWN → ∀ {x} → SM x → SN x
isWN∧isSM→isSN : R isWN → R isSM → R isSN
```

4.4 SN 推出 WN 的一个充分条件

强正规化蕴含弱正规化这一看似显而易见的命题实际上无法证明，这或许令人惊讶。事实上，文件 `ARS/Examples.agda` 表明，假设这一蕴含对自然数上的所有关系都成立，就可以推出排中律。同时，对于满足以下条件的所有关系，这一蕴含均成立：1）可判定；2）有限分支。文件 `ARS/Implications.agda` 将其形式化为

```
dec∧FB→SN⊆WN : R isDec → R isFB → SN ⊆ WN
```

由于所有标准项重写系统和 λ 演算实际上都满足这些条件，我们可以看到，对于它们所诱导的抽象重写系统，这一蕴含成立。

5 良基性

抽象归约 $\to_R$ 是强正规化的，当且仅当其逆关系是良基的。在本节中，我们将比较若干种良基性的定义。为简明起见，我们只考虑这些概念的全局版本。

XX:12 Agda 中抽象重写的形式化

5.1 良基性的定义

在本节余下部分，固定 $R \subseteq A \times A$。我们将 Rxy 记作 $(x,\ y) \in R$。在抽象重写系统的语境中，读者应将其理解为逆关系，并将 Rxy 解释为表示一个归约步 $y \to_R x$。对于 $P \subseteq A$，令 $\overline{P} = \{x \in A \mid x \notin P\}$ 表示 P 的补集。

▶ **定义 26**。

1. 若对所有 $y \in A$，Ryx 都蕴含 y 可达，则称 $x \in A$ 可达。这是一个归纳定义，其基础情形是满足对所有 y 都有 $\neg Ryx$ 的那些 x 。*2.* 若对每个 $x \in A$，$(\forall y.Ryx \to y \in P)$ 都蕴含 $x \in P$，则称 $P \subseteq A$ 是归纳的。*3.* 若 $x \in P$，且对于所有 y，Ryx 都蕴含 $y \notin P$，则称 $x \in A$ 是 P-极小的。*4.* 若对于每个 $x \in P$，都存在一个 $y \in \overline{A}$，使得 Ryx 且 $y \in P$，则称 $P \subseteq A$ 是余归纳的。

上述概念引出了几种不同的良基性定义，如下所示。

▶**定义 27**。

- R 是**可达良基的**，若每个元素都是可达的。R 是**归纳良基的**，若每个归纳谓词都成立。
- R 是**余归纳良基的**，若每个余归纳谓词都成立。R 是**按极小性良基的**，若非空子集都有极小元素。R 是**按极小性良基的，对于 $\neg\neg$-闭谓词**，若非空的 $\neg\neg$-闭子集都有极小元素。R 是**序列良基的**，若每个序列都包含一个它不递减的下标。
-
-

$(\mathbf{WFacc})\ \forall x \in A.\ x$ 是可达的$(\mathbf{WFind})\ \forall P \subseteq A.\ P$ 是归纳的 $\Rightarrow \forall x \in A.\ x \in P$
$(\mathbf{WFcor})\ \forall P \subseteq A.\ P$ 是余归纳的 $\Rightarrow \forall x \in A.\ x \in P(\mathbf{WFmin})$
$\forall P \subseteq A.\ P \neq \varnothing \Rightarrow \exists x \in A.\ x$ 是 P-极小的$(\mathbf{WFminDNE})$
$\forall P \subseteq A.\ P \neq \varnothing \Rightarrow \overline{P} \subseteq \overline{P} \Rightarrow \exists x \in A.\ x$ 是 P-极小的$(\mathbf{WFseq})\ \forall s : \mathbb{N} \to A\ \exists k :$
$\mathbb{N}.\ \neg\, R\,(s\,(k+1))\,(s\,k)$ $=$

我们已在 `WFDefinitions.agda` 中形式化了上述良基性的定义。

在经典逻辑下，上述良基性概念彼此等价。在构造性逻辑下，只有前两个等价（因此下文不再讨论 **WFind** ）；其余定义则不等价。此外，即使是单个经典定义，也可以按照不同的逻辑强度赋予构造性解释。形式化其余蕴含关系，使我们能够准确找出经典逻辑具体用在何处，以及真正需要多少经典逻辑。

例如，从“每个非空子集都有一个极小元”推出“每个元素都可达”，是一个运用经典反证法的证明。（假设存在一个不可达元素，根据假设，必定存在这样一个极小元素，而这会立即导致矛盾：每个极小元都是可达的。）

在构造性逻辑下，这个论证只能证明：如果每个非空子集都有一个极小元，那么没有元素是不可达的。也就是说，所证明的全称命题是：每个 $x \in A$ 都不是非可达的。要进一步推出 x 确实可达，通常无法在构造性逻辑下证明，因此必须将其作为附加假设，才能恢复完整的蕴含关系。

另一方面，假设可达性对 $\neg\neg$ 封闭，现在可以弱化另一个假设，只要求 $\neg\neg$ 封闭的子集存在极小元素。这是

一项重要限制，在某些情况下或许可以得到验证。相比之下，原始假设（断言所有谓词都存在极小元素）通常等价于每个子集都是可判定的。文件 `WFCounters.agda` 表明，假设自然数上的标准严格序满足完全极小性假设**WFmin**，会推出每个 $\mathbb{N}$ 的子集都可判定（这显然与有效语义不相容）；而其限制**WFminDNE**则意味着每个子集弱可判定，即对所有 x，均满足 $\neg\neg P(x) \vee \neg P(X)$。

也可以考虑证明“直至 $\neg\neg$”的良基性，即断言每个元素都是 $\neg\neg$-可达的，每个序列都 $\neg\neg$-包含一个非递减指标，对每个非空子集都 $\neg\neg$-存在一个极小元，等等。这些较弱的概念在下面的定义中形式化，并在相应的文件 `WFWeakDefinitions.agda`中给出。

▶**定义 28**。

- R 是弱可达良基，如果每个元素都是 $\neg\neg$-可达的：R 是弱余归纳良基，如果每个余归纳谓词在双重否定意义下对所有元素成立：R 是弱极小性良基，如果对每个非空子集 P 都 $\neg\neg$-存在 P 中的一个极小元：R 是弱极小性良基，对于 $\neg\neg$-闭谓词，如果对每个非空$\neg\neg$-闭子集 P 都 $\neg\neg$-存在 P 中的一个极小元：R 是弱序列良基，如果没有序列是 R-递减的：

(**WFacc**$\neg\neg$)	$\forall x \in A.\ \neg\neg(x \ \textit{is accessible})$
(**WFcor**$\neg\neg$)	$\forall P \subseteq A.\ P \ \textit{is coreductive} \Rightarrow \forall x \in A.\ \neg\neg(x \in P)$
(**WFmin**$\neg\neg$)	$\forall P \subseteq A.\ P \neq \varnothing \Rightarrow \neg\neg\ (\exists x \in A.\ x \ \textit{is } P\textit{-minimal})$
(**WFminDNE**$\neg\neg$)	$\forall P \subseteq A.\ P \neq \varnothing \Rightarrow \overline{\overline{P}} \subseteq P \Rightarrow \neg\neg\ (\exists x \in A.\ x \ \textit{is } P\textit{-minimal})$
(**WFseq–**)	$\forall s : \mathbb{N} \to A.\ \neg(s \ \textit{is } R\textit{-decreasing})$

5.2 各定义之间的蕴含关系

上述所有定义之间的逻辑关系总结于图 2 中。图中的所有蕴含关系都已在 Agda 代码库的 `WellFounded` 子目录中形式化。从一个节点指向另一个节点且没有标签的箭头表示直接的逻辑蕴含，可在不添加任何假设的情况下构造性地证明。带有标签的箭头则表示要确立该蕴含关系所需的额外（半）经典假设。这些假设的定义见图 3。

我们将**WFacc** 视为良基关系的典范构造性定义（因此图中特别强调了它）。在抽象重写系统的情形下，这相当于用归约到范式的方式归纳地定义强正规化。从图中可以看出，只有在满足两个 强经典条件时，**WFseq--** （Terese [8] 使用的定义）才蕴含它。这两个条件是**accDNE** （强正规化的双重否定闭包）和**accCor** （与可计算的持续约简策略的存在有关）。

图 2 展示了良基性的构造性概念之间的逻辑关系网络确实相当丰富。对这些概念取 $\neg\neg$-闭包，能够减少部分复杂性，但并非全部。例如，两个极小性原理变得等价。与此同时，从顺序定义的弱版本推出（弱）可达版本，仍然需要假设**accCor**；这可以描述为可达性定义的构造性逆否命题。在重写系统的语境中

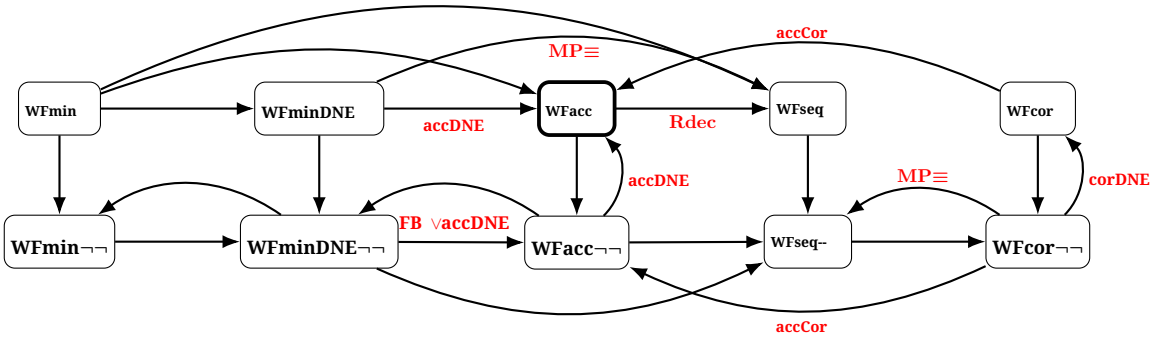


图 2 良基性概念之间的逻辑关系

性质	定义	Agda 代码
Rdec	R 是可判定的	∀ {x} {y} → EM (R x y)
FB	R 是有限分支的	∀（a : A） →Σ[xs ∈ ListA]（∀b → Rab → b ∈ List xs）
MP≡	序列是 ¬¬-闭的	∀ (f : ℕ → A) → ¬¬Closed(λx →Σ[k ∈ ℕ](f k ≡ x))
corDNE	余归纳性是 ¬¬-闭的	∀(P: $\mathcal{P}$A) → R-coreductive P → ¬¬Closed P
accDNE	可达性是 ¬¬-闭的	¬¬Closed (R -accessible)
accCor	可达性是余归纳的	R -coreductive (R -accessible)

图 3 良基性图中使用的经典性质

，该条件断言存在一个函数，它将任意一个不 强正规化的元素映射到它的一个一步归约，且该归约具有相同的性质。假设**accCor**，所有良基性概念在 ¬¬ 下构造性等价。

从图中还可以得出以下观察结果。1. 从 WFacc 到 WFseq 的蕴含要求底层关系是可判定的。在 ¬¬-翻译的一侧，这一假设不再必要。2. WFmin 是一个非常强的假设，并且无条件蕴含 WFseq。较为合理的 WFminDNE 则要求每个序列的像都是 ¬¬-闭的，才能完成这一蕴含。此条件可视为马尔可夫原理的一个特例（假设 A上的相等是可判定的）。3. 在 ¬¬-一侧，WFmin¬¬ 与 WFminDNE¬¬ 等价。两者都直接由 WFacc¬¬蕴含，且这三者都直接蕴含 WFseq--。4. 所有定义都蕴含 WFcor¬¬。这个最弱的定义在假设“可达性的构造性逆否命题”accCor 成立时，蕴含 WFacc¬¬ 。在这一假设下，也可以从 WFcor 推导出 WFacc。5. 如果关系是有限分支的（FB），或者 R-可达性是 ¬¬-闭的（accDNE），那么 WFminDNE¬¬ 直接蕴含 WFacc¬¬。（值得注意的是，只要关系 R 是可判定的，且R-可达性是弱可判定的，FB 就蕴含 accCor；这一点在 `Coreductive.agda`中得到了证明。）

5.3 良基性与重写

有人可能会想知道，上述各种定义之间的差异对重写理论是否重要。特别是，对于给定的重写系统，图 3 中列出的性质实际上有多常成立，

以至于可以从较弱的陈述（例如**WFminDNE**¬¬）推断出其终止性（**WFacc**）？

一方面，[7] 证明了一个一般性结果，其大意是：**accDNE** 对一大类可定义关系在元理论意义上成立。具体而言，若 R 是自然数上的原始递归谓词，且**WFacc**¬¬(R)可在高阶算术中得到证明，则在高阶算术中也必定存在**WFacc**(R)的直觉主义证明。因此，几乎不可能通过一个显式反例来区分这些概念。

另一方面，其中某些性质可以通过表明它们之间的蕴含会得出一个“构造性禁忌”来加以区分：即一个已知在没有额外经典假设的情况下不可证的命题。例如，良基关系的典型例子是自然数上的严格序 < 。通常的归纳法很容易对 <确立**WFacc** 。与此同时，文件 `WFCounters.agda` 表明，对 < 断言**WFmin** 蕴含自然数的每一个子集都是可判定的。由于这在没有经典公理的情况下显然无法在 Agda 中证明，该论证也表明从**WFacc** 到 **WFmin** 的蕴含同样不可证。

这些区别会对重写理论产生影响。例如，回想一下，$x \in \mathrm{SN}_R$ 当且仅当 x 是 R-可达的，其中 R 是 R的逆关系。因此，如果 R 是良基的，$R \models$**强正规化**。然而，实际计算给定 x 的范式，需要以较强的形式判定其是否为 R-极小：$\forall x.(\exists y.x \longrightarrow_R y) \vee (\forall y.x \not\longrightarrow_R y)$（注意，仅判定 x是否为范式，只需判定较弱的析取命题 $\neg\neg($ $\exists y.x \longrightarrow_R y) \vee (\forall y.x \not\longrightarrow_R y)$。）

另一方面，**WFmin** 显然蕴含每个 $x \in A$ 都是弱正规化的。事实上，给定 $x \in A$，考虑谓词 $P_x(y) = x \longrightarrow_R^* y$。该谓词非空，因为空归约给出了 $P_x(x)$的证明。令 n 为一个 P_x-极小元。那么$x \longrightarrow_R^* n$。此外，对于任意 y，若 $n \longrightarrow_R y$，则 $\neg(P_x(y))$。但由于从 x 出发的归约对向下封闭，不存在这样的 y，因此 n 是 x的范式。这表明，蕴含式**WFacc** → **WFmin** 不可证这一结论也适用于强正规化和弱正规化。

6 一个应用：λ项的性质

为了说明我们这项工作的普遍适用性，我们展示它如何与更大项目中的另一个组件——λ演算——结合。

我们对 λ 演算的形式化基于 λ项的所谓嵌套类型 编码。[9] [5]。（我们从 Håkon Gylterud 那里了解到这种编码；它似乎在 Haskell 社区中相当流行。）这种表示的关键在于，不要将 λ项视为一个集合（类型），而要将其视为一个函子（一个类型索引族）：

```
data Λ (X : Set) : Set where
var : X → ΛX
app : ΛX → ΛX → ΛX
abs : Λ(↑X) → ΛX
```

↑ 类型构造子表示提升单子 $\uparrow X = X + 1$。它出现在 `abs` 构造子中 Λ 的参数内，使得该类型成为非齐次类型：集合 A 上 λ项的类型取决于其他集合上项的类型。

这种编码的优点包括以下几点：与 de Bruijn 指标相关的提升/降阶引理获得了自然的范畴形式；这种编码启发出一种通用的高阶归纳组合子，足以推导出所有标准结果，例如代换引理；

XX：16 Agda中抽象重写的形式化

该框架足以对经典λ演算理论进行形式化。当前库的内容包括关于归约（标准化和合流性）的标准结果、基于将不可解项等同起来的不可判定的λ理论的定义（$\mathcal{H}$ 和$\mathcal{H}\omega$），以及对一项未发表的二循环刻画结果的形式化；该结果基于此前与 Jan Willem Klop 和 Joerg Endrullis 的合作。

作为 ARS库 如何助力这项工作的一个例子，我们通过抽象重写系统理论得到了以下基本推论：**强正规化 → 弱正规化：** Agda能够有效地计算任意强正规化项的范式。**合流性 → 唯一范式性$^{=}$：** 不同的范式不可β可转换（等式一致性）。$\mathtt{decNF}_{\beta}$**：** 可以判定给定项是否为范式。**良基性：** 图 2 中展示的大多数性质对于 λ项而言是等价的。尽管这些事实都很简单，本可以直接证明，但它们能够作为一般理论的推论得出，恰恰说明了这一切的意义：这些事实本就不该需要单独证明，而现在确实无需再证明了。

这项应用另一个值得注意的方面是，Λ 类型族的异质性丝毫没有妨碍对抽象重写系统进行经典的 Terese风格的形式化。例如，范式可判定性的证明源于这样一个事实：单步β归约是可判定且有限分支的。这些概念是在固定的论域 A 上定义的——尽管单个 λ抽象需要考虑不同集合上的项之间的归约，因此需要在新的论域中实例化抽象重写系统理论。这表明，Agda的模块系统确实与其类型系统配合得很好。令人意外的是，用于命名空间管理的设计特性在关联高度通用性的结果时竟如此有用。

7 结论与后续工作

总之，我们已将 [8]中所呈现的抽象重写系统理论的基本结果形式化。我们的研究表明，该理论的很大一部分对于编程语言理论中通常遇到的大多数语言都可以做到可计算。在这一工作过程中，我们发现经典定理往往假设了超出必要的条件。通过在每个证明中力求优化假设，我们对标准的终止性与合流性判据获得了边际改进。我们还识别出若干新的抽象重写系统性质，并确立了它们与标准概念之间的关系。这使我们更清晰地从结构上理解不同性质如何相互作用，从而有助于对抽象重写系统理论作出更精细的表述。

我们还分析了在构造性元理论中精确表述良基关系概念时的微妙之处。我们识别出若干经典原理，它们是将这些概念的不同变体联系起来所必需的。

我们的工作为进一步扩展奠定了基础。它使标准的抽象重写系统结果可立即用于项重写系统和类型化λ演算的形式化。进一步工作的一个潜在方向是开发一种工具，用于将外部证明器的终止性证书导入 Agda。例如，实现从年度国际终止性竞赛所用格式的转换 [11] 将使未来的开发能够使用最先进的自动终止性证明器。

参考文献

1 Peter Aczel 等。同伦类型论：数学的单价基础。单价基础项目，高等研究院，2013年。2 Agda开发团队。Agda 标准库：列表。https://agda.github.io/agda-stdlib/v2.2/Agda.Builtin.List.html，2025年。访问日期：2025年7月8日。3 Agda开发团队。Agda 标准库：重写。 https://agda.github.io/agda-stdlib/v1.7.3/Relation.Binary.Rewriting.html，2025年。访问日期：2025年11月5日。4 Thorsten Altenkirch、Paolo Capriotti、Gabe Dijkstra、Nicolai Kraus 与 Fredrik Nordvall Forsberg。商归纳-归纳类型。载于软件科学与计算结构基础国际会议，第293–310页。施普林格国际出版公司，查姆，2018年。5 Thorsten Altenkirch 与 Bernhard Reus。使用广义归纳类型的λ项单子表示。载于计算机科学逻辑，第13届国际研讨会（CSL '99），《计算机科学讲义》第1683卷，第453–468页。施普林格，1999年。doi:10.1007/3-540-48168-0_32。6 Sam Arkle 与 Andrew Polonsky。阿巴拉契亚州立大学编程语言基础。https://github.com/DrPolonsky/FPLA，2026年。访问日期：2026年3月11日。7 Stefano Berardi。带高阶概括的经典算术与直觉主义算术在归纳良基性上一致。载于 Stephan Kreutzer 编，第24届EACSL计算机科学逻辑年会（CSL 2015），莱布尼茨国际信息学论文集（LIPIcs）第41卷，第343–358页，德国达格施图尔，2015年。Dagstuhl 城堡 – 莱布尼茨信息学中心。URL： https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CSL.2015.343， doi:10.4230/LIPIcs.CSL.2015.343。8 Marc Bezem、Jan Willem Klop 与 Roel de Vrijer。项重写系统。剑桥大学出版社，2003年。9 Richard S. Bird 与 Ross Paterson。作为嵌套数据类型的 de Bruijn记法。《函数式编程杂志》，9(1)：77–91，1999年。 doi:10.1017/S0956796899003366。10 Frédéric Blanqui 与 Adam Koprowski。Color：关于良基重写关系的 Coq库及其在终止性证书自动验证中的应用。《计算机科学中的数学结构》，21(4)：827–859，2011年。 doi:10.1017/S0960129511000120。11 年度国际终止性竞赛。 https://termination-portal.org/wiki/Termination_Competition。访问日期：2025年11月13日。12 Thierry Coquand、Simon Huber 与 Anders Mörtberg。论立方类型论中的高阶归纳类型。载于第33届ACM/IEEE计算机科学逻辑年会论文集，第255–264页，2018年。13 André L. Galdino 与 Mauricio Ayala-Rincón。项重写系统的 PVS理论。《理论计算机科学电子札记》，247：67–83，2009年。第三届逻辑与语义框架及其应用研讨会论文集（LSFA 2008）。URL： https://www.sciencedirect.com/science/article/pii/S1571066109002722, doi:10.1016/j.entcs.2009.07.049。14 Christian Sternagel 与 René Thiemann。抽象重写。形式化证明档案，2010年6月。 https://isa-afp.org/entries/Abstract-Rewriting.html，形式化证明开发。