arXiv:2304.03175v1 [cs.PL] 6 Apr 2023

统一线性性与依赖性分析

PRITAM CHOUDHURY,宾夕法尼亚大学，美国

线性性分析与依赖性分析是计算机科学中若干应用的关键，尤其是在资源管理和信息流控制方面。这两类分析的共同点在于，它们都需要对至少两个相互作用受到约束的不同世界进行建模。具体而言，典型的线性性分析会对非线性世界和线性世界进行建模，并规定非线性世界中的推导不能使用来自线性世界的假设；典型的依赖性分析则会对低安全级世界和高安全级世界进行建模，并规定高安全级世界中的信息绝不会泄漏到低安全级世界。尽管线性性分析与依赖性分析解决的问题相似，但它们采用的方法却各不相同。在线性性分析中，类型系统采用 Girard 线性逻辑中的余单子指数模态；在依赖性分析中，类型系统采用 Moggi 的计算元语言中的单子模态。由于方法上的差异，统一这两种分析虽然在理论和实践上都很有价值，却并非易事。

值得庆幸的是，随着分级上下文类型系统的最新进展，人们认识到，线性性分析和依赖分析可以从同一视角来考察。然而，现有的分级上下文类型系统尚不足以统一线性性分析和依赖分析。其问题在于，尽管这些系统的线性性分析具有一般性，依赖分析却受到限制，这主要是因为它们采用的分级模态具有余单子性，而不具有单子性。本文针对这一局限，对现有分级上下文类型系统进行系统扩展，使分级模态同时具有余单子性和单子性。通过这一扩展，我们得以将线性性分析与一般的依赖分析统一起来。我们提出统一的线性依赖演算 LDC，它在任意纯类型系统中使用同一机制分析线性性和依赖关系。我们证明，LDC 是一种通用的线性与依赖演算系统，因为它涵盖了针对这两种分析各自的标准演算系统。

CCS 概念：• **计算理论** → **线性逻辑**；**类型理论**；• **安全与隐私** → 形式化安全模型。

附加关键词与短语：分级类型系统、依赖类型、堆语义

1 引言

类型系统将我们对程序正确性的直觉形式化：形式化之后，正确的程序就是类型良好的程序。然而，我们对正确性的直觉会随应用场景而变化。例如，在分布式系统中，正确性意味着多个用户不能同时拥有对某个文件的写入权限；在安全系统中，正确性意味着公共用户无法读取机密文件。为了形式化这些以及其他类似的正确性概念，我们采用线性类型系统和依赖类型系统。这些类型系统有着广泛的应用。

线性类型系统擅长管理资源使用。它们能够推理状态，并支持编译器优化，例如对内存位置进行原地更新[Wadler 1990]。它们可用于精细的内存管理 [Chirimar 等人 2000；Fluet 等人 2006]。通过会话类型，它们能够保证分布式系统中不会发生死锁 [Caires 等人 2016]。鉴于其效用，线性类型已应用于多种编程语言，例如 Haskell [Bernardy 等人 2018],Granule [Orchard 等人 2019], Idris 2 [Brady 2021]。

依赖类型系统擅长控制信息流。以安全类型系统的形式 [Heintze 和 Riecke, 1998；Volpano 等人 1996], 它们保证低安全级别输出不依赖于高安全级别输入。以绑定时间类型系统的形式 [Davies, 2017；Gomard 和 Jones 1991],它们保证早期绑定表达式不依赖于晚期绑定表达式。依赖类型系统在实用编程语言中相当常见。例如，元编程语言 MetaOcaml [Calcagno 等人 2003] 基于

依赖类型系统的 Davies [2017]。用于确保安全信息流的 Java 扩展 Jif 基于 Myers [1999]。

尽管线性类型系统和依赖类型系统服务于不同的目的，但它们本质上处理的是同一个抽象问题。这个问题是对至少两个遵循给定约束并相互作用的不同世界进行建模。例如：典型的线性类型系统需要对非线性世界和线性世界进行建模，并规定非线性世界中的推导不能使用来自线性世界的假设；典型的依赖类型系统需要对低安全级世界和高安全级世界进行建模，并规定来自高安全级世界的信息绝不会泄漏到低安全级世界。这种根本上的相似性表明，线性与依赖分析可以得到统一。

这种统一具有诸多益处。首先，从理论角度来看，它将统一用于线性性分析的标准演算系统 [Barber1996；Benton 1994] 与用于依赖分析的标准演算系统 [Abadi 等人 1999；Shikuma and Igarashi 2006]。其次，从实践角度来看，它将允许程序员对线性性分析和依赖分析使用同一种类型系统。目前，程序员为此目的使用不同的类型系统。例如，Haskell 程序员使用 Linear Haskell 扩展 [Bernardy 等人 2018] 进行线性性分析，并使用 LIO 库 [Stefan 等人 2017]进行依赖分析。第三，它将实现这两种分析的组合。组合分析比单独进行的各项分析更强大，因为它允许任意组合使用约束和流约束。例如，在组合分析中，一段数据可能同时具有线性性和私密性。

统一线性类型系统与依赖类型系统虽然十分理想，但并非易事。这是因为这些类型系统采用了不同的方法来施加各自的约束。线性类型系统 [Abramsky1993；Barber 1996；Benton 等人 1993；Turner 和 Wadler1999] 采用取自线性逻辑 余单子性的 指数模态 !， [Girard 1987]，以便在基础线性世界中操作非线性资源。另一方面，依赖类型系统[Abadi 等人 1999；Heintze 和 Riecke 1998；Volpano 等人 1996] 采用取自计算元语言 单子性的 模态T， [Moggi 1991]，，以便在基础低安全级世界中操作高安全级值。

模态 ! 和 T 的行为不同。例如：在依赖演算中，不存在类型为 $T\,\mathsf{Bool} \to \mathsf{Bool}$ 的非恒定函数，因为任何此类函数都会泄露信息。然而，在线性演算中存在一个非恒定函数，$\emptyset \vdash \lambda x.\mathsf{derelict}\, x : !\,\mathsf{Bool} \to \mathsf{Bool}$，因为非线性资源可以按线性方式使用。接下来，在依赖演算中，$x : \mathsf{Bool} \vdash_\eta x : T\,\mathsf{Bool}$ 是一个有效的类型判断，因为信息可以从低安全级世界流向高安全级世界。然而，在线性演算中，$x : \mathsf{Bool} \vdash\!! x : !\,\mathsf{Bool}$ 不是一个有效的类型判断，因为非线性资源无法使用线性假设。由于这些差异，很长一段时间以来，线性性分析和依赖性分析一直彼此独立地进行。

然而，随着分级上下文类型系统的最新进展 [Abel 和 Bernardy 2020年；Brunel 等人 2014；Ghica 和 Smith 2014； Orchard等人 2019年； Petricek等人 2014], 人们已经认识到，这两种分析可以通过同一个视角来理解。分级上下文类型系统是参数化于预序半环或类似抽象代数结构之上的类型系统。这些类型系统通过对参数化结构进行不同实例化，可以执行多种多样的用法分析。换句话说，它们并不局限于线性/非线性用法分析，还可以分析其他形式的用法，例如无用法、仿射使用、有界用法等。因此，分级上下文类型系统比线性类型系统更加一般。现在，为了理解分级上下文类型系统如何分析依赖关系，我们需要从用法分析的角度考察模态 ！ 和 T之间的相似性。假设基础世界是线性的，单子模态 T可以理解为表示无用法，因为来自

一个“单子世界”不能在该世界之外使用。这种理解将模态 !（表示无限制使用）和 T（表示不使用）纳入同一框架，也让我们能够从一个视角看待上文提到的它们行为上的差异。

然而，现有分级上下文类型系统的问题在于，尽管其线性性分析具有通用性，但其依赖分析却受到限制。这些系统无法捕捉依赖分析的若干方面。我们将在下一节详细讨论这些方面。这一缺陷背后的主要原因是，分级上下文类型系统是为分析共效应而设计的，即分析程序如何依赖其上下文。共效应包括线性性（单次使用）、无关性（不使用）等。依赖关系的表现则更像一种效应。具体来说：低级和高级安全级别计算可以分别被视为纯计算和具有效应的计算。像依赖关系这样的效应，无法被现有的分级上下文类型系统很好地捕捉，而这些系统基本上属于共效应演算。

本文表明，通过系统地扩展现有的分级上下文类型系统，我们可以将其用于一般的依赖关系与线性性分析。我们设计了一种演算系统LDC，能够使用相同的机制，同时分析诸如线性性的反效应和诸如依赖关系的效应。LDC以任意纯类型系统为参数，并涵盖用于线性与依赖分析的标准演算系统。我们通过堆语义证明了 LDC 中的线性性与依赖关系分析是正确的。

总而言之，我们做出了以下贡献：

- 我们提出了一种语言 LDC，它以任意纯类型系统为参数，使用同一种机制分析线性性与依赖关系。
- 我们证明，LDC 涵盖了用于分析线性性与依赖关系的标准演算系统，例如线性非线性 λ演算（Benton） [1994],、DCC（Abadi 等人 [1999], 提出的封装演算）（Shikuma 和 Igarashi） [2006],等。
- 我们证明，LDC 中线性性与依赖分析的正确性均可由该演算的可靠性定理推出。
- 我们证明，LDC 能够执行组合线性与依赖分析。

2 挑战与解决方案

2.1 依赖分析：显著方面

在上一节中，我们讨论了关于低安全级世界和高安全级世界的依赖分析。这种分析可以扩展到任意（有限）数量的世界，并考虑它们之间的依赖约束。Denning [1976]观察到，对世界施加依赖约束会形成一种格结构。回忆一下，格 $\mathrm{L} = (L, \sqsubseteq)$ 是一个偏序集，其中每一对元素都有一个最小上界，也称为合并，记作 $\sqcup$；还有一个最大下界，也称为交，记作 $\sqcap$。如果 L 是有限的，那么它具有一个顶元素和一个底元素，分别记作 $\top$ 和 $\bot$，使得 $\bot \sqsubseteq \ell \sqsubseteq \top$，对于所有 $\ell \in \mathrm{L}$。现在，为了给出一个依赖格的示例，考虑以下世界集合：一个低安全级世界 L、两个彼此不共享信息的中安全级世界M_1 和M_2，以及一个高安全级世界 H。这些约束可以用一个菱形格来建模，即 $\mathrm{L}_\diamond$，其中 $\mathrm{L} \sqsubseteq \mathrm{M}_1 \sqsubseteq \mathrm{H}$ 且$\mathrm{L} \sqsubseteq \mathrm{M}_2 \sqsubseteq \mathrm{H}$，其含义是，任何违背这一格序的信息流都是非法的。更一般地，对于任意格L，给定级别 ℓ_1、$\ell_2 \in \mathrm{L}$，我们有：$\ell_1 \sqsubseteq \ell_2$ 当且仅当信息可以从 ℓ_1 流向 ℓ_2。

依赖类型系统 [Abadi 等人 1999；Shikuma 和 Igarashi 2006] 建立在这一信息流格模型之上。这些类型系统使用取自抽象依赖格的标签，对 Moggi 的计算元语言中的单子模态 T进行分级，$\mathcal{L}$。分级背后的理念是，对于任意 $\ell \in \mathcal{L}$，$T_\ell\, A$ 表示 ℓ-安全的 A 类型项。形象地说，

T_ℓ A 表示类型为 A 的项，但这些项被封装在 ℓ-安全的盒子中，只有具备 ℓ-安全许可才能打开。例如， T_H Bool 是高安全级别布尔值的类型，只有具备高安全级别许可才能观察这些布尔值。现在，如果依赖类型系统能够确保：每当 $\neg(\ell_1 \sqsubseteq \ell_2)$ 时， ℓ_1-安全的输入不会影响 ℓ_2-安全的输出，那么就可以称该系统是健全的。这一原则也可以表述为：每当 $\neg(\ell_1 \sqsubseteq \ell_2)$ 时，通过观察 ℓ_2 处的输出，无法获得关于 ℓ_1 中输入的任何信息。这就是著名的非干扰原则，依赖类型系统的力量正源于此。举一个具体例子，该原则的推论是：给定安全格 $\mathrm{L}_\circ$，任何类型为 T_H Bool $\to T_\mathrm{L}$ Bool 的函数都是常函数。

现在，我们来考察分级模态的本质， T_ℓ。在分析依赖关系时，应当能够在类型 $T_\ell\ T_\ell$ A 和 T_ℓ A 之间自由转换，因为将一个项包裹在一串 ℓ-安全盒中，与将其包裹在单个 ℓ-安全盒中是相同的。从 T_ℓ A 转换到 $T_\ell\ T_\ell$ A 很容易，因为只需添加一个额外的包装器即可。但从 $T_\ell\ T_\ell$ A 转换到 T_ℓ A 则需要进行一些考虑，因为这涉及解包。为处理这种情况，可以利用模态的性质。模态具有单子性，支持标准的合并操作： $T_{\ell_1}\ T_{\ell_2}$ A $\to T_{\ell_1 \sqcup \ell_2}$ A，其中 $\ell_1, \ell_2 \in \mathrm{L}_\circ$。通过这一合并操作，可以从 $T_\ell\ T_\ell$ A 转换到 T_ℓ A。现在，如果依赖标签对应于效应，那么合并操作就对应于计算这些效应的并集。正如计算并集在效应演算中十分重要，拥有合并操作在依赖演算中也十分重要。在本节后文中，我们将看到，现有的分级上下文类型系统无法推导出合并操作。这严重限制了这些类型系统中的依赖分析。

在进一步讨论之前，我们想指出依赖分析中另一个有时会被忽视的重要方面。这涉及如何处理被 T_H包装的函数。考虑类型： T_H (T_H 布尔值 $\to$ 布尔值)。这种类型的值应该是什么？ DCC [Abadi 等人 1999] 会回答：只有 $\eta_\mathrm{H}(\lambda \mathrm{x}\,.$真值) 和 $\eta_\mathrm{H}(\lambda \mathrm{x}\,.$假值)。但封装演算 [Shikuma 和 Igarashi 2006] 会回答：$\mathrm{seal}_\mathrm{H}(\lambda \mathrm{x}\,.$真值)、$\mathrm{seal}_\mathrm{H}(\lambda \mathrm{x}\,.$假值)、$\mathrm{seal}_\mathrm{H}(\lambda \mathrm{x}\,.\mathrm{unseal}_\mathrm{H}\ \mathrm{x})$ 和 $\mathrm{seal}_\mathrm{H}(\lambda \mathrm{x}\,.\mathrm{not}\ (\mathrm{unseal}_\mathrm{H}\ \mathrm{x}))$。这种差异源于这样一个事实：在封装演算中，函数 T_H 布尔值 $\to$ 布尔值若被 T_H包装，就可能返回高安全级值；而在 DCC 中，它必须始终返回常量输出。在这方面，封装演算比 DCC 更通用，因为它不会限制任何函数返回高安全级值，只要该函数本身被高安全级别标签 H 包装即可。请注意，对于会终止的计算，封装演算涵盖 DCC；事实上，正如上文所见，它比 DCC 更通用。

封装演算比 DCC 更一般之处，在于其类型判断的形式。DCC 使用传统形式的类型判断，而封装演算使用如下形式的带标签类型判断：

$$x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n \vdash b :^\ell B, \tag{1}$$

其中 $\ell \in \mathrm{L}_\circ$。这个判断的直观含义是：假设 x_i 的类型为 A_i，那么 b 是一个类型为 B 的 ℓ-安全项，对于 $i = 1, 2, \ldots, n$ 均如此。带标签的类型判断的主要优点是，它能使安全值的密封和解封过程更加顺畅，下面的规则便体现了这一点：

$$\textsc{SC-Seal}\quad \frac{\Gamma \vdash a :^{\ell \sqcup \ell_0} A}{\Gamma \vdash \mathrm{seal}_{\ell_0}\ a :^\ell T_{\ell_0}\ A} \qquad \textsc{SC-Unseal}\quad \frac{\Gamma \vdash a :^\ell T_{\ell_0}\ A \qquad \ell_0 \sqsubseteq \ell}{\Gamma \vdash \mathrm{unseal}_{\ell_0}\ a :^\ell A}$$

另一方面，DCC 需要采用一种非标准的绑定规则，并辅以一个保护性判断，才能解封安全值。如上所述，封装演算的带标签的类型判断也使任意函数都能返回高安全级别输出，前提是函数本身被封装在高安全级别标签之下。鉴于这些优点，我们采用封装



(a) $\mathcal{Q}$Lin 中的排序　　(b) $\mathcal{Q}$Aff 中的排序

图 1. 跟踪线性和仿射用法的排序

演算作为依赖分析的模型。顺带一提，尽管封装演算不允许非终止计算，但我们的语言允许。有关封装演算与 DCC 更详尽的比较，请参阅 Choudhury [2022]。

2.2 分级上下文类型系统：主要特点

近年来，分级上下文类型系统 [Abel 和 Bernardy 2020；Atkey 2018；Brunel 等人 2014；Choudhury 等人 2021； Gaboardi 等人 2016； Ghica 和 Smith 2014； McBride 2016； Moon 等人 2021；Orchard 等人 2019； Petricek 等人 2014] 已成功用于推理程序中的资源使用。分级上下文类型系统源自有界线性逻辑 [Girard 等人 1992]，它将线性逻辑的“命题即资源”原则加以改造，以刻画资源受限计算。为实现这一目标，与线性逻辑相比，有界线性逻辑对资源用法作出了更细致的区分。分级上下文类型系统借鉴有界线性逻辑，拓宽了用法分析的性质和范围，使其超越传统线性类型系统 [Barber 1996； Benton 1994]。

分级上下文类型系统的强大功能和灵活性，源于它们以抽象预序半环或类似结构作为参数，而这种结构表示一种资源代数。回顾一下，半环（Q, +, ·, 0, 1）是一个集合 Q，带有两个二元运算符： +（加法）和 ·（乘法），以及各自的单位元 0 和 1；其中加法满足交换律和结合律，乘法满足结合律并对加法满足分配律，且 0 是乘法的吸收元。预序半环 $\mathsf{Q} = (Q, +, \cdot\, 0, 1, <\!:)$ 是一个半环（Q, +, ·, 0, 1），带有一个与二元运算相容的二元预序关系 $<\!:$；也就是说，若 $\mathsf{q}_1 <\!: \mathsf{q}_2$，，则 $\mathsf{q} + \mathsf{q}_1 <\!: \mathsf{q} + \mathsf{q}_2$，且 $\mathsf{q} \cdot \mathsf{q}_1 <\!: \mathsf{q} \cdot \mathsf{q}_2$，并且 $\mathsf{q}_1 \cdot \mathsf{q} <\!: \mathsf{q}_2 \cdot \mathsf{q}$，对所有 $q \in Q$均成立。通过改变参数 Q，分级上下文类型系统可以执行多种分析。其中，我们特别关注线性性和仿射性这两种分析。这两种分析所用的预序半环分别记为 Q_{Lin} 和 Q_{Aff}，二者都包含 3 个元素：0、1 和ω，且满足 $1 + 1 = \omega + 1 = 1 + \omega = \omega + \omega = \omega$和 $\omega \cdot\ \omega = \omega$。然而，$\mathsf{Q}_{\text{Lin}}$ 和 Q_{Aff} 的序关系不同，如图 1 所示。由于序关系不同，1 在 Q_{Lin} 中表示线性使用，而在 Q_{Aff}中表示仿射使用。不过，在 Q_{Lin} 和 Q_{Aff}中，0 都表示不使用，而 ω 表示无限制使用。

分级上下文类型系统使用参数化预序半环 Q 的元素来对!模态进行分级。分级背后的思想是，对于任意 $\mathsf{q} \in \mathsf{Q}$，$!_{\mathsf{q}}A$ 表示类型为 A的q-使用项。例如，当 Q设为Q_{Lin}时，类型 $!_1 A$ 表示必须恰好使用一次的类型为A 的项。类似地，当Q 设为 Q_{Aff} 时，类型 $!_1 A$ 表示至多使用一次的项。然而，在这两种情况下，类型 $!_\omega A$ 都表示可以不受限制地使用的项。现在，如果一个分级上下文类型系统能够正确地核算使用情况，从而确保使用公平性，则可以称其是健全的。使用公平性意味着不存在任意复制或丢弃资源。举一个具体例子，在一个由 Q_{Lin} 参数化的分级上下文多态类型系统中，类型 $\forall\alpha.!_1\alpha \to !_1\alpha \times !_1\alpha$和 $\forall\alpha.!_1\alpha \to$单元类型应当是不可居住的。

分级上下文类型系统的关键特征是：它使用参数化结构中的元素，为类型判断的上下文进行分级。由预序半环 Q 参数化的分级上下文类型系统中的典型类型判断如下：

$$x_1 :^{q_1} A_1, x_2 :^{q_2} A_2, \ldots, x_n :^{q_n} A_n \vdash b : B, \tag{2}$$

其中 $q_i \in \mathsf{Q}$。请注意，分级上下文 Γ 可以分解为非分级上下文 $\lfloor\Gamma\rfloor$ 和等级向量 Γ。分级上下文 $x_1 :^{q_1} A_1,\ x_2 :^{q_2} A_2,\ \ldots\ldots,\ x_n :^{q_n} A_n$ 的直观含义与标准上下文 $x_1 : !_{q_1} A_1,\ x_2 : !_{q_2} A_2,\ \ldots\ldots,\ x_n : !_{q_n} A_n$ 相同。然而，使用分级上下文代替标准上下文背后有一个非常充分的理由：通过将预序半环的相应运算提升到上下文层面，可以轻松地在分级上下文上定义上下文运算。例如，对于分级上下文 Γ_1 和 Γ_2，给定 $\lfloor\Gamma_1\rfloor = \lfloor\Gamma_2\rfloor$，上下文加法运算 $\Gamma_1 + \Gamma_2$ 通过对等级进行逐点加法来定义，而上下文乘法运算 $q \cdot \Gamma$ 则通过将 Γ 中的每个等级与 q 进行前乘来定义。这些上下文运算有助于在分级上下文类型系统中顺畅地核算资源使用。下面给出的 $!q$ 的引入规则和消去规则说明了这一点。

$$\textsc{GC-ExpIntro}\ \frac{\Gamma \vdash a : A}{q \cdot \Gamma \vdash !_q\, a :!_q A} \qquad \textsc{GC-ExpElim}\ \frac{\Gamma_1 \vdash a :!_q A \quad \Gamma_2, x :^q A \vdash b : B \quad \lfloor\Gamma_1\rfloor = \lfloor\Gamma_2\rfloor}{\Gamma_1 + \Gamma_2 \vdash \text{let } !_q\, x \text{ be } a \text{ in } b : B}$$

接下来，我们考虑分级模态的性质，$!_q$。此分级模态在 $(Q, \cdot, 1)$ [Fujii 2019年; Katsumata 2018], 这意味着，可以为此模态导出标准的提取函数和分叉函数：

$$\begin{aligned}\text{extract} &:= \lambda x.\text{let } !_1\, y \text{ be } x \text{ in } y \ :\ !_1 A \to A\\ \text{fork}^{q_1,q_2} &:= \lambda x.\text{let } !_{q_1 \cdot q_2}\, y \text{ be } x \text{ in } !_{q_1}\, !_{q_2}\, y \ :\ !_{q_1 \cdot q_2} A \to !_{q_1}\, !_{q_2} A\end{aligned}$$

这里，A 是任意类型，而 q_1,$\mathsf{q}_2 \in Q$。这种模态的余单子性对于支持用法分析至关重要。举例来说，对于提升$!_\omega$A $\to !_\omega !_\omega$A，需要 $\text{fork}^{\omega,\omega}$。尽管分叉函数对于用法分析是必需的，但对应的合并函数却并非如此。因此，分级上下文类型系统不会推导出合并函数。事实上，这些类型系统通常无法推导出合并函数，也就是说，在这些类型系统中不存在任何类型为 $!_{q_1} !_{q_2}$A $\to !_{q_1 \cdot q_2}$A 的可推导一般函数。附录 A 中的命题A.1给出了这一论断的模型论证明。分级上下文类型系统的这一缺陷对于用法分析无关紧要，但却限制了依赖分析，下一节将对此进行说明。

2.3 分级上下文类型系统中依赖分析的局限性

现有的分级上下文类型系统在依赖关系分析方面存在局限。下面我们将详细说明原因。

（1）分级上下文类型系统由预序半环参数化。然而，依赖关系分析由格参数化。预序半环和格都是具有两个二元运算符及一个二元序关系的代数结构。不过，这些结构之间的一个关键区别在于：半环需要确保一个运算符（乘法）对另一个运算符（加法）具有分配律，而格则不需要满足任何此类性质。事实上，存在两个运算符彼此都不满足分配律的格。此类格中最简单的是 M_3 和 N_5 [Birkhoff 1967],，它们都有 5 个元素，其序关系如图 2所示。在 M_3中，合并和交彼此不满足分配律：



(a) 格 M_3 (b) 格 N_5

图 2. 非分配格示例

$$(\ell_1 \sqcup \ell_2) \sqcap (\ell_3 \sqcup \ell_2) = \top \neq \ell_2 = (\ell_1 \sqcap \ell_3) \sqcup \ell_2$$
$$(\ell_1 \sqcap \ell_2) \sqcup (\ell_3 \sqcap \ell_2) = \bot \neq \ell_2 = (\ell_1 \sqcup \ell_3) \sqcap \ell_2.$$

N_5 也是如此：

$$(\ell_2 \sqcup \ell_1) \sqcap (\ell_3 \sqcup \ell_1) = \ell_3 \neq \ell_1 = (\ell_2 \sqcap \ell_3) \sqcup \ell_1$$
$$(\ell_1 \sqcap \ell_3) \sqcup (\ell_2 \sqcap \ell_3) = \ell_1 \neq \ell_3 = (\ell_1 \sqcup \ell_2) \sqcap \ell_3.$$

因此，任意格都不能视为预序半环。然而，分配格，即合并与交彼此满足分配律的格，可以视为预序半环，其中乘法、加法和序分别由合并、交和格序给出。因此，现有的分级上下文类型系统有望对分配格进行依赖分析。然而，我们认为这是一个局限，因为我们看不出将依赖分析限制在分配格内有任何原则性依据。

(2) 依赖分析需要合并运算符，正如第 2.1 节所讨论的。然而，分级上下文类型系统无法推导出合并运算符，正如第 2.2 节所讨论的。解决这一问题的一种临时办法是向类型系统中显式添加一个合并运算符，如下所示：

$$\text{GC-Join}\ \frac{\Gamma \vdash a :!_{q_1} !_{q_2} A}{\Gamma \vdash \mathrm{join}^{q_1,q_2}\ a :!_{q_1 \cdot q_2} A}$$

然而，由于以下原因，这种解决方案并不十分令人满意：

- 这个合并运算符给操作语义带来了问题，尤其是在按名称调用归约中。要理解其中的原因，请考虑以下问题：在按名称调用演算中，合并^{q_1,q_2} b 应如何归约？为了归约以合并开头的项，可以考虑以下规则：

$$\text{Step-JoinLeft}\ \frac{\vdash a \rightsquigarrow a'}{\vdash \mathrm{join}^{q_1,q_2}\ a \rightsquigarrow \mathrm{join}^{q_1,q_2}\ a'} \qquad \text{步-合并Beta}\ \frac{}{\vdash \mathrm{join}^{q_1,q_2}\ !_{q_1} !_{q_2} a \rightsquigarrow !_{q_1 \cdot q_2} a}$$

接下来，为了归约 join^{q_1,q_2} b，首先要将 b 归约为一个值，然后再尝试应用规则 Step-JoinBeta。但请注意，即使对于类型良好的项，这种尝试也可能不会成功。这是因为在按名称调用演算中，无论 a′ 本身是否为值，$!_q$a′ 都是一个值。因此，join^{q_1,q_2}b 的归约可能会停在类似 $\mathrm{join}^{q_1,q_2} !_{q_1}$ b′ 这样的项上，其中 b′ 并非以 $!_{q_2}$ 开头。在这种情况下，项 $\mathrm{join}^{q_1,q_2} !_{q_1}$b′ 无法通过规则 Step-JoinBeta继续归约。鉴于这种情况，为了维持类型可靠性，就不得不将 $\mathrm{join}^{q_1,q_2} !_{q_1}$b′ 这样的项声明为值，从而允许演算中出现缺乏原则依据的值。

- 此外，引入这一 合并 运算符会破坏类型系统的对称性，因为规则 GC-Join既不是 ! 的引入规则，也不是消去规则。由于对称性遭到破坏，便无法以有原则的方式确保 合并 是 分叉的逆运算；而分叉已可在类型系统中推导出来，如第 2.2节所示。因此，这种 合并运算符同样不利于类型系统的等式理论。

  鉴于这些缺点，我们认为，在分级上下文类型系统中添加临时合并运算符以实现依赖分析并非正确的做法。因此，我们避免采用这种方法。

(3) 接下来，正如第 2.1 节所述，我们希望有一种依赖演算，使任意函数只要自身被包裹在高安全级标签下，就能返回高安全级值。现有的分级上下文类型系统不允许这样做。例如，在一个以格 $\mathsf{L} \sqsubseteq \mathsf{H}$ 为参数的分级上下文类型系统中，类型 $!_{\mathsf{H}}(!_{\mathsf{H}}\mathsf{Bool} \to \mathsf{Bool})$ 只包含两个不同的项：$!_{\mathsf{H}}(\lambda x.\mathsf{true})$ 和 $!_{\mathsf{H}}(\lambda x.\mathsf{false})$。换句话说，在分级上下文类型系统中，不存在与封装演算中的 $\mathsf{seal}_{\mathsf{H}}(\lambda x.\mathsf{unseal}_{\mathsf{H}} x)$ 和 $\mathsf{seal}_{\mathsf{H}}(\lambda x.\mathsf{not}(\mathsf{unseal}_{\mathsf{H}} x))$ 相对应的项。

因此，我们看到，现有的分级上下文类型系统中的依赖分析存在若干障碍。这促使我们寻求统一线性性和依赖分析的其他解决方案。

2.4 迈向消解

Choudhury 等人的最新工作Choudhury 等人 [2022] 指出了一种统一线性性和依赖分析的可能方法。Choudhury 等人 [2022]提出了DDC$^{\top}$，这是纯类型系统中用于一般依赖分析的类型系统。DDC$^{\top}$虽然类似于现有的分级上下文类型系统，但避免了上述所有局限。DDC$^{\top}$ 由任意格参数化，并涵盖封装演算。此外，DDC$^{\top}$ 能够在依赖类型环境中分析依赖关系，例如依赖类型程序中的运行时无关性。DDC$^{\top}$与分级上下文类型系统之间的关键区别在于其类型判断的形式。DDC$^{\top}$除了对类型判断的上下文进行分级之外，还在判断上添加标签，如下所示：

$$x_1 :^{\ell_1} A_1, x_2 :^{\ell_2} A_2, \ldots, x_n :^{\ell_n} A_n \vdash b :^{\ell} B, \tag{3}$$

其中 ℓ_i，ℓ 是参数化格中的元素。检验符右侧的标签（3）表示观察者级别。在分级上下文类型系统中，观察者级别固定为 1。改变观察者级别所增加的灵活性使 DDC$^{\top}$ 执行一般的依赖分析。

然而，DDC$^{\top}$ 无法进行线性性分析。因此，统一线性性与依赖性分析的问题仍然悬而未决。不过，我们可以从 DDC$^{\top}$ 中汲取灵感，看看当我们允许分级上下文类型系统通过以下形式的类型判断改变观察者级别时会发生什么：

$$x_1 :^{q_1} A_1, x_2 :^{q_2} A_2, \ldots, x_n :^{q_n} A_n \vdash b :^{q} B, \tag{4}$$

其中 q_i，$q \in Q$。这个类型判断本质上是封装演算的类型判断与现有分级上下文类型系统的类型判断的融合，分别如（1）和（2）所示。这似乎是一个良好的开端。然而，一条已知的障碍正等待着我们。 Atkey [2018]证明了，使用上述判断形式并以一个任意 半环为参数化的类型系统不允许替换。这个障碍令人气馁，但我们发现这并非绝路。我们发现，尽管在某些半环上替换不可容许，但在若干个我们感兴趣的预序半环上，替换实际上是可容许的。特别地，替换在以下半环上是可容许的

类型 A, B, C ::= 单元类型 | $^{r}\mathrm{A} \to \mathrm{B}$ | $^{r}\mathrm{A}_1 \times \mathrm{A}_2$ | $\mathrm{A}_1 + \mathrm{A}_2$

项 a, b, c ::= x | $\lambda^{r}\mathrm{x}\text{: A.b}$ | b a^{r}

| 单元类型 | let$_{q_0}$ 单元类型 be a in b | $(\mathrm{a}_1{}^{r}, \mathrm{a}_2)$ | let$_{q_0}$(x ^{r}, y) be a in b

| inj$_1$ a_1 | inj$_2$ a_2 | case 表达式$_{q_0}$ a 的 $\mathrm{x}_1.\mathrm{b}_1$; $\mathrm{x}_2.\mathrm{b}_2$

上下文 Γ ::= $\emptyset$ | Γ,x :^{q} A

图 3. LDC($\mathrm{Q}_{\mathbb{N}}$) 的文法（简单版本）

用于追踪线性性和仿射性的标准预序半环，即分别为 $\mathrm{Q}_{\mathrm{Lin}}$ 和 $\mathrm{Q}_{\mathrm{Aff}}$。

因此，我们终于找到了一种统一线性性和依赖性分析的方法。在我们的线性依赖演算 LDC 中，我们使用（4）中所示形式的类型判断。对于线性性及其他用法分析，我们将 LDC 参数化为某些预序半环，并将在后文逐步描述这些半环。对于依赖分析，我们将 LDC 参数化为一个任意的格。对于结合的线性性与依赖分析，我们将 LDC 参数化为用于各项独立分析的结构的笛卡尔积。

现在，LDC 可以在任意纯类型系统中分析线性性和依赖关系。不过，演算的一些关键思想在简单类型环境中最容易解释。因此，我们首先介绍 LDC 的简单类型版本，之后再将其推广到纯类型系统版本。

3 简单类型系统中的线性与依赖分析

3.1 线性分析类型系统

首先，我们将分析精确用量和有界用法。我们将在第 ω节中，将无限制使用（记作 ω）加入我们的演算中。6。精确用量可以通过 $\mathbb{N}_{=}$进行分析，即带离散序的自然数半环。有界用法可以通过 $\mathbb{N}_{\geq}$进行分析，即带自然数降序的自然数半环。 $\mathbb{N}_{\geq}$ 中的序如下所示： $\ldots <: 4 <: 3 <: 2 <: 1 <: 0$。序关系存在差异的原因是，在有界用法分析中，资源可以被丢弃。但请注意，在这两种分析中，资源都不会被复制。

接下来，我们介绍以这两种预序半环为参数的 LDC。需要明确指出时，我们将以代数结构 AS 为参数的 LDC 称为 LDC(AS)。代数结构 AS 可以是预序半环、格，或它们的笛卡尔积（用于组合分析时）。

现在，令 $\mathrm{Q}_{\mathbb{N}}$ 在$\{\mathbb{N}_{=}, \mathbb{N}_{\geq}\}$ 上变化。LDC($\mathrm{Q}_{\mathbb{N}}$) 的文法见图 3。该演算具有函数类型、积类型、和类型以及Unit 类型。积类型对应于线性逻辑的乘法合取；和类型对应于加法析取。注意，函数类型和积类型都标注了等级。类型 $^{r}\mathrm{A} \to \mathrm{B}$和 $^{r}\mathrm{A}_1 \times \mathrm{A}_2$本质上分别是 $(!_r \mathrm{A}) \to \mathrm{B}$ 和 $(!_r \mathrm{A}_1) \times \mathrm{A}_2$。我们本可以使用未标注的函数类型和积类型，即分别使用$\mathrm{A} \to \mathrm{B}$和$\mathrm{A}_1 \times \mathrm{A}_2$，以及分级指数模态类型 $!_r\mathrm{A}$，来代替 $^{r}\mathrm{A} \to \mathrm{B}$ 和 $^{r}\mathrm{A}_1 \times \mathrm{A}_2$,。然而，我们选择以这种方式呈现，因为它很容易推广到纯类型系统设置。对于项，我们有与这些类型相对应的引入形式和消去形式。项也标注了等级，用于追踪它们所使用的资源。上下文中的假设也会连同它们的允许用量一起出现。

$\boxed{\Gamma \vdash a :^q A}$ （简单版本）

ST-Var

$$\frac{}{0 \cdot \Gamma_1, x :^q A, 0 \cdot \Gamma_2 \vdash x :^q A}$$

ST-Lam

$$\frac{\Gamma, x :^{q \cdot r} A \vdash b :^q B}{\Gamma \vdash \lambda^r x{:}A.b :^q A \to^r B}$$

ST-应用

$$\frac{\Gamma_1 \vdash b :^q A \to^r B \quad \Gamma_2 \vdash a :^{q \cdot r} A \quad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash b\ a^r :^q B}$$

ST-Pair

$$\frac{\Gamma_1 \vdash a_1 :^{q \cdot r} A_1 \quad \Gamma_2 \vdash a_2 :^q A_2 \quad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash (a_1{}^r, a_2) :^q A_1 \times^r A_2}$$

ST-LetPair

$$\frac{\Gamma_1 \vdash a :^{q \cdot q_0} A_1 \times^r A_2 \quad \Gamma_2, x :^{q \cdot q_0 \cdot r} A_1, y :^{q \cdot q_0} A_2 \vdash b :^q B \quad q_0 <: 1 \quad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash \text{令}^{q_0}(x^r, y) \text{ 为 } a, \text{ 在 } b \text{ 中} :^q B}$$

ST-Unit

$$\frac{}{0 \cdot \Gamma \vdash \text{unit} :^q \text{单元类型}}$$

ST-LetUnit

$$\frac{\Gamma_1 \vdash a :^{q \cdot q_0} \text{单元类型} \quad \Gamma_2 \vdash b :^q B \quad q_0 <: 1 \quad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash \text{let}^{q_0} \text{ 单元类型是 } a \text{ 在 } b :^q B}$$

ST-Inj1

$$\frac{\Gamma \vdash a_1 :^q A_1}{\Gamma \vdash \text{inj}_1\ a_1 :^q A_1 + A_2}$$

ST-Inj2

$$\frac{\Gamma \vdash a_2 :^q A_2}{\Gamma \vdash \text{inj}_2\ a_2 :^q A_1 + A_2}$$

ST-Case

$$\frac{\Gamma_1 \vdash a :^{q \cdot q_0} A_1 + A_2 \quad \Gamma_2, x_1 :^{q \cdot q_0} A_1 \vdash b_1 :^q B \quad \Gamma_2, x_2 :^{q \cdot q_0} A_2 \vdash b_2 :^q B \quad q_0 <: 1 \quad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash \text{case 表达式}^{q_0} a \text{ of } x_1.b_1 ;\ x_2.b_2 :^q B}$$

ST-SubL

$$\frac{\Gamma' \vdash a :^q A \qquad \Gamma <: \Gamma'}{\Gamma \vdash a :^q A}$$

ST-SubR

$$\frac{\Gamma \vdash a :^q A \qquad q <: q'}{\Gamma \vdash a :^{q'} A}$$

图 4. LDC($Q_{\mathbb{N}}$) 的类型规则

接下来，我们来看类型系统。类型系统见图 4。需要注意以下几点：

- 分级上下文 Γ 本质上由底层的非分级上下文（记作 $\Delta := \lfloor \Gamma \rfloor$）以及相应的等级向量（记作 Γ）组合而成。
- 对于满足 $\lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor = \Delta$ 的上下文 Γ_1 和 Γ_2，我们定义 $\Gamma := \Gamma_1 + \Gamma_2$ 如下：$\lfloor \Gamma \rfloor = \Delta$ 和 $\Gamma = \Gamma_1 + \Gamma_2$（逐点向量加法）。
- 对于上下文 Γ_0 和等级 q，我们定义 $\Gamma := q \cdot \Gamma_0$ 如下：$\lfloor \Gamma \rfloor = \lfloor \Gamma_0 \rfloor$ 和 $\Gamma = \overline{q} \cdot \Gamma_0$（标量乘法）。

- 对于满足 $\lfloor \Gamma \rfloor = \lfloor \Gamma' \rfloor$ 的上下文 Γ 和 Γ'，当且仅当 $\Gamma <: \Gamma'$（逐点序）时，我们称 $\Gamma <: \Gamma'$。

- 对于任意上下文 Γ，我们都隐含地假设，没有两个假设为同一个变量指定类型。

类型判断 $x_1 :^{q_1} A_1, x_2 :^{q_2} A_2, \ldots, x_n :^{q_n} A_n \vdash a :^{q} A$ 可以直观地理解为：可以使用 q_i 个类型为 A_i 的 x_i，产生 q 个类型为 A 的 a，其中 $i = 1, 2, \ldots, n$。基于这种理解，让我们来看看类型规则。

大多数规则都符合预期。需要注意的是，消去规则 ST-LetPair、ST-LetUnit 和 ST-Case 有一个侧条件 $q_0 <: 1$。设置此条件的原因是，要归约这些消去形式中的任意一种，我们都必须先归约项 a，这意味着无论如何，我们都需要用于归约至少一个 a 副本的资源。我们可以将 q_0 设为 1，但允许任意 $q_0 <: 1$ 会使这些规则更灵活。例如，得益于这种灵活性，在规则 ST-LetPair 中，b 可以使用 $q \cdot q_0$ 个 y 的副本，而不只是 q 个 y 的副本。另一个值得注意之处是，规则 ST-SubL 和 ST-SubR 如何帮助丢弃 LDC($\mathbb{N}_{\geq}$) 中的资源。

现在，我们来看几个 LDC($\mathbb{N}_{\geq}$) 中可推导项和不可推导项的例子。（为简洁起见，我们省略 λs 中的定义域类型。）

$\emptyset \vdash \lambda^1 x.x :^{1\ 1} A \to A$ but $\emptyset \nvdash \lambda^0 x.x :^{1\ 0} A \to A$

$\emptyset \vdash \lambda^1 x.x :^{2\ 1} A \to A$ but $\emptyset \nvdash \lambda^1 x.(x^2, \mathrm{unit}) :^{1\ 1} A \to {}^2A \times \mathrm{Unit}$

$\emptyset \vdash \lambda^1 x.\mathrm{let}_1\ (x_1{}^1, x_2)\ \mathrm{be}\ x\ \mathrm{in}\ x_1 :^{1\ 1} ({}^1A \times A) \to A$ but $\emptyset \nvdash \lambda^1 x.(x^1, x) :^{1\ 1} A \to {}^1A \times A$

仔细考察后，我们发现上述不可推导项无法公平地使用资源。考虑这些项的类型：^{0} A → A、 ^{1}A → ^{2}A × 单元类型以及 ^{1}A → ^{1}A × A。只有在资源可以被复制的情况下，这些类型才可能有值。由于我们不允许复制，它们实际上是无值的。然而，由于LDC($\mathbb{N}_{\geq}$) 允许丢弃资源，类型 ^{1}(^{1}A × A) → A 实际上是有值的。

注意，为了生成任意项的 0 个副本，我们不需要任何资源。因此，在 0 世界中，只要某个类型去掉注释后的对应类型是可居住的，任何带注释的类型都是可居住的。不过，资源在 0 世界中毫无意义。换言之，判断 $\Gamma \vdash a :^0 A$ 所传达的信息，并不比其对应的标准 λ-演算版本更多。

接下来，我们将考察 LDC($\mathbb{Q}_{\mathbb{N}}$) 的操作语义和元理论。

3.2 线性分析的元理论

首先，我们考虑一些句法性质。该演算满足下面陈述的乘法引理。这个引理表明，如果我们需要 Γ 资源来生成 q 个 a 的副本，那么我们需要 $r_0 \cdot \Gamma$ 资源来生成 $r_0 \cdot q$ 个 a 的副本。

引理 3.1（乘法）。若 $\Gamma \vdash a :^{q} A$，则 $r_0 \cdot \Gamma \vdash a :^{r_0 \cdot q} A$。

请注意，我们没有在论文正文中给出引理和定理的证明，但相关证明见附录。

演算也满足如下所述的因子分解引理。该引理指出，如果一个上下文Γ 产生 q 份 a，那么 Γ 可以拆分为 q 个部分，其中每个部分都产生 1 份 a。我们需要前提条件 $q \neq 0$，因为资源在 0 世界中没有意义，但在所有其他世界中都有意义，尤其是在 1 世界中。

引理 3.2（因子分解）。如果 $\Gamma \vdash a :^{q} A$ 且 $q \neq 0$，则存在 Γ' 使得 $\Gamma' \vdash a :^{1} A$且 $\Gamma <: q \cdot \Gamma'$。

利用上面的两个引理，我们可以证明如下所述的分裂引理。该引理表明，如果我们拥有资源 Γ，以产生 $q_1 + q_2$ 份 a，那么我们可以将 Γ 拆分为两部分，Γ_1 和 Γ_2，使得 Γ_1 和 Γ_2 可以分别产生 q_1 和 q_2 份 a。Atkey [2018]指出，在针对某些半环进行参数化时，分裂引理并不适用于与我们的类型系统类似的类型系统。然而，对于我们用于参数化 LDC 的预序半环，分裂引理成立，即 $\mathbb{N}_{=}$ 和 $\mathbb{N}_{\geq}$。

⊢一个 ⇝ 一个′

（节选）

应用贝塔步骤

$$\frac{}{\vdash (\lambda^{r} x{:}A.b)\ a^{r} \rightsquigarrow b\{a/x\}}$$

Step-LetPairBeta

$$\frac{}{\vdash 令_{q_0}(x^{r},\ y) 为 (a_1{}^{r},\ a_2),\ 则\ b \rightsquigarrow b\{a_1/x\}\{a_2/y\}}$$

图 5. LDC($Q_{\mathbb{N}}$) 的小步归约

引理 3.3（拆分）。 如果 $\Gamma \vdash a :^{q_1+q_2} A$，则存在 Γ_1 和 Γ_2 使得 $\Gamma_1 \vdash a :^{q_1} A$ 且$\Gamma_2 \vdash a :^{q_2} A$ 且 $\Gamma = \Gamma_1 + \Gamma_2$。

接下来，我们来看弱化和替换。对于弱化，我们添加一个额外假设，其允许用量为 0。对于替换，我们需要确保变量的允许用量与替代项可用副本的数量相匹配。请注意，替换之后，该项需要合并后的资源。

引理 3.4（弱化）。如果 $\Gamma_1, \Gamma_2 \vdash a :^{q} A$，那么 Γ_1，$z :^{0} C$ $\Gamma_2 \vdash a :^{q} A$。

引理 3.5（替换）。如果 Γ_1，$z :^{r_0} C$ $\Gamma_2 \vdash a :^{q} A$ 且 $\Gamma \vdash c :^{r_0} C$ 并且 $\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$，则$\Gamma_1 + \Gamma$，$\Gamma_2 \vdash a\{c/z\} :^{q} A$。

现在，我们考虑该语言的操作语义。我们的操作语义是一种按名称调用的小步语义。除了 β-规则（如图 5 所示）之外，所有步进规则都是标准的。这些规则确保引入形式中的等级与消去形式中的等级相匹配。借助这一操作语义，我们的语言具备标准的类型可靠性性质。

定理 3.6（保持性）。 如果 $\Gamma \vdash a :^{q} A$且 $\vdash a \rightsquigarrow a'$，则 $\Gamma \vdash a' :^{q} A$。

定理3.7（进展）。若 $\emptyset \vdash a :^{q} A$，则 a 要么是一个值，要么存在a' 使得$\vdash a \rightsquigarrow a'$。

现在，请考虑一个事实：在标准的小步语义中，我们不会替换缩约项中的自由变量。因此，这样的语义无法真正对程序的资源使用进行建模。基于环境的语义会将项的自由变量替换为环境中的值，因此更适合对程序的资源使用进行建模。在第 4节中，我们将为这一演算介绍一种基于环境的语义，并展示该类型系统能够正确地处理资源使用。不过现在，我们继续讨论依赖分析。

3.3 用于依赖分析的类型系统

令 $L = (L, \sqcap, \sqcup, \top, \bot, \sqsubseteq)$ 为任意格。（严格来说，格不一定具有顶元素和底元素，但我们总可以添加它们。）我们使用 ℓ和 m 表示 L的元素。现在，将 +、·、0、1和 <: 分别解释为 ⊓、⊔、 ⊤、 ⊥ 和 ⊑，并使用 q 和 r 表示L 的元素，我们在图4所示的类型系统中得到一个依赖演算。在图6中，我们展示了该类型系统中一些选定的规则，仅改变了记法。

现在，类型系统以 L 作为参数，而不再使用 $Q_{\mathbb{N}}$。我们按照与对应项 $q \cdot \Gamma$、 $\Gamma_1 + \Gamma_2$ 和 $\Gamma <$相同的方式定义 $\ell \sqcup \Gamma$、 $\Gamma_1 \sqcap \Gamma_2$ 和 $\Gamma \sqsubseteq \Gamma'$：分别为 Γ'。类型判断 $x_1 :^{\ell_1} A_1, x_2 :^{\ell_2} A_2, \ldots, x_n :^{\ell_n} A_n \vdash b :^{\ell} B$可读作：假设 x_i 在 ℓ_i 可观测，则 b 在 ℓ 可观测，其中 $i = 1, 2, \ldots, n$。按照这种理解，我们来考察一些类型规则。

规则 ST-VarD 符合预期。规则 ST-LamD 很有意思。类型 $^{m}A \to B$ 包含一些函数，这些函数接受的参数是 m-安全的，即只能在 m 及更高级别上观察到。

$\boxed{\Gamma \vdash \mathsf{a} :^{\ell} \mathsf{A}}$ （选定规则）

$$\text{ST-VarD}\ \frac{}{\top \sqcup \Gamma_1, \mathsf{x} :^{\ell} \mathsf{A}\ \top \sqcup \Gamma_2 \vdash \mathsf{x} :^{\ell} \mathsf{A}} \qquad \text{ST-LamD}\ \frac{\Gamma, \mathsf{x} :^{\ell \sqcup m} \mathsf{A} \vdash \mathsf{b} :^{\ell} \mathsf{B}}{\Gamma \vdash \lambda^{m} \mathsf{x} : \mathsf{A}.\mathsf{b} :^{\ell}\ ^{m}\mathsf{A} \to \mathsf{B}} \qquad \text{ST-AppD}\ \frac{\Gamma_1 \vdash \mathsf{b} :^{\ell}\ ^{m}\mathsf{A} \to \mathsf{B} \quad \Gamma_2 \vdash \mathsf{a} :^{\ell \sqcup m} \mathsf{A} \quad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 \sqcap \Gamma_2 \vdash \mathsf{b}\ \mathsf{a}^{m} :^{\ell} \mathsf{B}}$$

$$\text{ST-PairD}\ \frac{\Gamma_1 \vdash \mathsf{a}_1 :^{\ell \sqcup m} \mathsf{A}_1 \quad \Gamma_2 \vdash \mathsf{a}_2 :^{\ell} \mathsf{A}_2 \quad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 \sqcap \Gamma_2 \vdash (\mathsf{a}_1{}^{m},\ \mathsf{a}_2) :^{\ell}\ ^{m}\mathsf{A}_1 \times \mathsf{A}_2} \qquad \text{ST-LetPairD}\ \frac{\Gamma_1 \vdash \mathsf{a} :^{\ell \sqcup \ell_0}\ ^{m}\mathsf{A}_1 \times \mathsf{A}_2 \quad \Gamma_2, \mathsf{x} :^{\ell \sqcup \ell_0 \sqcup m} \mathsf{A}_1, \mathsf{y} :^{\ell \sqcup \ell_0} \mathsf{A}_2 \vdash \mathsf{b} :^{\ell} \mathsf{B} \quad \ell_0 \sqsubseteq \bot \quad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 \sqcap \Gamma_2 \vdash \text{令}\ell_0\ (\mathsf{x}^{m},\ \mathsf{y})\ \text{为}\ \mathsf{a},\ \text{在}\ \mathsf{b}\ \text{中}:\ ^{\ell}\mathsf{B}}$$

$$\text{ST-SubLD}\ \frac{\Gamma' \vdash \mathsf{a} :\ ^{\ell}\mathsf{A} \qquad \Gamma \sqsubseteq \Gamma'}{\Gamma \vdash \mathsf{a} :\ ^{\ell}\mathsf{A}} \qquad \text{ST-SubRD}\ \frac{\Gamma \vdash \mathsf{a} :\ ^{\ell}\mathsf{A} \qquad \ell \sqsubseteq \ell'}{\Gamma \vdash \mathsf{a} :^{\ell'}\ \mathsf{A}}$$

图 6。LDC(L) 的类型规则

因此，检查此类函数的函数体时，需要将参数的可观察性级别设置为至少 m。

相反，在规则 ST-AppD中，参数 a 只需在 m 或更高层级上可观测。注意，结论判断中的上下文由逐点取检查函数和参数的上下文的交而形成。这确保前提中可观测的子项不会在结论中变得不可观测。

规则 ST-WPairD 展示了如何将一个安全项装入一个可能不安全的世界中。如果 a_1是 m-安全的，那么我们可以在级别 ℓ释放它，但只有在将其放入一个可在 m 及更高级别打开的盒子之后。这类似于封装演算的模态类型 $T_m\ A$ 本质上$^m A$ × 单元类型保护信息的方式，正如我们在规则 SC-Seal中所看到的那样。

反之，规则 ST-LetPairD确保一个 m-安全盒只能在 m 及更高层级打开。注意，在此规则中，侧条件 $\ell_0 \sqsubseteq \bot$ 强制 ℓ_0 为 $\bot$，因为 $\bot$ 是底元素。因此，规则 ST-LetPairD可以在将 ℓ_0 设为 $\bot$ 后简化。然而，我们以这种方式呈现该规则，以强调 LDC($\mathcal{Q}_{\mathbb{N}}$) 和 LDC($\mathcal{L}$) 之间的相似性。

接下来，我们来看几个 LDC($\mathcal{L}_{\diamond}$) 中可推导项和不可推导项的例子，其中 $\mathcal{L}_{\diamond}$是菱形格（参见第 2.1 节）。

令 Bool := Unit +Unit，且 true := inj_1 unit，false := inj_2 unit。那么，

$$\emptyset \vdash \lambda^{\mathrm{L}} x.x :^{\mathrm{H}}\ ^{\mathrm{L}}\mathrm{Bool} \to \mathrm{Bool}\ \text{but}\ \emptyset \nvdash \lambda^{\mathrm{H}} x.x :^{\mathrm{L}}\ ^{\mathrm{H}}\mathrm{Bool} \to \mathrm{Bool}$$

$$\emptyset \vdash \lambda^{\mathrm{H}} x.\eta_{\mathrm{M}_1}\ \eta_{\mathrm{M}_2}\ x :^{\mathrm{L}}\ ^{\mathrm{H}}\mathrm{Bool} \to T_{\mathrm{M}_1}\ T_{\mathrm{M}_2}\ \mathrm{Bool}\ \text{but}\ x :^{\mathrm{H}}\ \mathrm{Bool} \nvdash x :^{\mathrm{M}_1}\ \mathrm{Bool}$$

这里，$T_m \mathsf{A} := {}^{m}\mathsf{A} \times \mathsf{Unit}$，且 $\eta_{\ell}\,\mathsf{a} := (\mathsf{a}^{\ell},\ \mathsf{unit})$。仔细观察可以发现，不可推导项违反了 $\mathrm{L}_{\diamond}$所建模的依赖关系约束。第一项将信息从 H 传递到 L，而第二项则将信息从 H 传递到 M_1。另一方面，可推导项遵守了 $\mathrm{L}_{\diamond}$所建模的依赖关系约束。第一项将信息从 L 传递到 H，而第二项则将 H 中的信息封装在一个嵌套于另一个 M_1 盒子中的 M_2 盒子里。

下面，我们给出在 LDC(L) 中见证标准合并和分叉操作的项。（注意，任何被擦除的注释都假定为 $\bot$。）回顾一下，现有的分级上下文类型系统

无法推导出这样的合并运算符。LDC(L) 虽然与这些类型系统相似，但由于其灵活的类型判断，可以做到这一点，而这种类型判断允许人们改变观察者级别。

$$\emptyset \vdash c_1 : T_{\ell_1}\ T_{\ell_2}\ A \to T_{\ell_1 \sqcup \ell_2}\ A \text{ and } \emptyset \vdash c_2 : T_{\ell_1 \sqcup \ell_2}\ A \to T_{\ell_1}\ T_{\ell_2}\ A \text{ where,}$$

$$c_1 := \lambda x.\eta_{\ell_1 \sqcup \ell_2}\ (\text{let } (y^{\ell_2}, _) \text{ be } (\text{let } (z^{\ell_1}, _) \text{ be } x \text{ in } z) \text{ in } y)$$

$$c_2 := \lambda x.\eta_{\ell_1}\ \eta_{\ell_2}\ (\text{let } (y^{\ell_1 \sqcup \ell_2}, _) \text{ be } x \text{ in } y)$$

这些项的推导见附录J。

接下来，我们考察 LDC(L) 的元理论。

3.4 依赖性分析的元理论

我们考察 LDC($\mathcal{Q}_{\mathbb{N}}$) 的依赖性对应性质，这些性质在第3.2节中给出。其中大多数性质对 LDC($\mathcal{L}$) 都成立。LDC($\mathcal{L}$) 满足乘法引理。该引理指出，我们总能同时提升上下文以及观察导出项的层级。

引理 3.8（乘法）。若 $\Gamma \vdash \mathsf{a} :^{\ell} \mathsf{A}$，则$\mathsf{m}_0 \sqcup \Gamma \vdash \mathsf{a} :^{m_0 \sqcup \ell} \mathsf{A}$。

拆分　　ng 引理也是真值。由于 $\sqcap$ 是幂等的，根据规则 ST-SubRD。

引理 3.9（拆分）。如果 $\Gamma \vdash a :^{\ell_1 \sqcap \ell_2} A$，则存在 Γ_1 和 Γ_2，使得 $\Gamma_1 \vdash a :^{\ell_1} A$ 以及$\Gamma_2 \vdash a :^{\ell_2} A$ 和 $\Gamma = \Gamma_1 \sqcap \Gamma_2$。

然而，因子分解引理在此并不成立，因为若它成立，就会导致安全信息泄露。来看具体原因，考虑以下 LDC($\mathsf{L}_{\diamond}$) 判断：$\emptyset \vdash \lambda \mathsf{x}.\mathsf{let}(\mathsf{y}^{\mathrm{H}}, _)\mathsf{bex\ iny} :^{\mathrm{H}} T_{\mathrm{H}}\ \mathsf{Bool} \to \mathsf{Bool}$。若因子分解引理成立，那么根据此判断，我们将得到：$\emptyset \vdash \lambda \mathsf{x}.\mathsf{let}\ (\mathsf{y}^{\mathrm{H}}, _)\ \mathsf{be\ x\ in\ y} :^{\mathrm{L}} T_{\mathrm{H}}\,\mathsf{Bool} \to \mathsf{Bool}$，这是一个从 $T_{\mathrm{H}}\,\mathsf{Bool}$ 到 Bool 的非恒定函数，处于 L 安全世界中，代表着安全信息的泄露。请注意，在 LDC($\mathsf{Q}_{\mathbb{N}}$) 中，证明分裂引理需要用到因子分解引理，而分裂引理又是证明替换所必需的。在 LDC(L) 中，分裂引理显然成立，无需任何因子分解引理。

接下来，我们考虑弱化和替换。弱化引理会在上下文中添加一个处于最高安全级别的额外假设。另一方面，替换引理会将一个处于 m_0 的假设替换为一个在 m_0 处推导出的项。

引理 3.10（弱化）。若 $\Gamma_1, \Gamma_2 \vdash \mathsf{a} :^{\ell} \mathsf{A}$，则 $\Gamma_1, \mathsf{z} :^{\top} \mathsf{C}\ \Gamma_2 \vdash \mathsf{a} :^{\ell} \mathsf{A}$。

引理 3.11（替换）。若 $\Gamma_1, \mathsf{z} :^{m_0} \mathsf{C}\ \Gamma_2 \vdash \mathsf{a} :^{\ell} \mathsf{A}$ 且 $\Gamma \vdash \mathsf{c} :^{m_0} \mathsf{C}$ 且 $\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$，则$\Gamma_1 \sqcap \Gamma, \Gamma_2 \vdash \mathsf{a}\{\mathsf{c/z}\} :^{\ell} \mathsf{A}$。

现在，LDC(L) 可以采用与 LDC($\mathsf{Q}_{\mathbb{N}}$) 相同的操作语义。对于这种操作语义，LDC(L) 具备标准的类型可靠性性质。

定理 3.12（保持性）。若 $\Gamma \vdash \mathsf{a} :^{\ell} \mathsf{A}$ 且 $\vdash \mathsf{a} \rightsquigarrow \mathsf{a}'$，则 $\Gamma \vdash \mathsf{a}' :^{\ell} \mathsf{A}$。

定理 3.13（进展）。若 $\emptyset \vdash a :^{\ell} A$，则要么 a 是一个值，要么存在 a' 使得$\vdash a \rightsquigarrow a'$。

类型可靠性不足以证明 LDC($\mathcal{L}$) 能正确分析依赖关系。为此，我们需要证明该演算通过非干扰测试，以确保 ℓ_1-输入的变化不会影响 ℓ_2-输出，只要 $\neg(\ell_1 \sqsubseteq \ell_2)$。在第 4 节中，我们将证明 LDC($\mathcal{L}$) 通过非干扰测试。不过在此之前，我们先讨论 LDC($\mathcal{L}$) 与封装演算之间的关系。

3.5 封装演算与LDC

封装演算 [Shikuma 和Igarashi 2006] 嵌入 LDC($\mathcal{L}$) 中。由于 LDC($\mathcal{L}$) 与 Choudhury等人 [2022]的 SDC 相同，因此我们不在此提供嵌入的详细信息。Choudhury 等人[2022]证明了封装演算嵌入到 SDC中。不过请注意，SDC 只能参数化于格，而不能参数化于有助于跟踪线性性的半环。

LDC(L) 与 SDC 之间的一个技术差异在于，SDC 的函数类型和积类型上没有安全性注释。不过，SDC 具有模态类型，T_ℓ A。LDC(L) 中带注释的函数类型和积类型，^{m}A $\to$B 以及 ^{m}A $\times$B，分别对应于 SDC 中的类型 T_mA $\to$B 和 T_mA $\times$B。反过来，SDC 的模态类型 T_m A 对应于 LDC(L) 中的 ^{m}A $\times$ 单元类型。

鉴于封装演算是一种可嵌入 LDC(L) 的通用依赖演算，我们得出结论：LDC(L) 也是一种通用依赖演算。此外，LDC($\mathrm{Q}_{\mathbb{N}}$) 可以分析资源使用。因此，通过针对适当的结构参数化该演算，LDC 可同时用于使用情况分析和依赖性分析。在下一节中，我们将通过该演算的堆语义证明这些分析的正确性。

4 LDC 的堆语义

LDC 对资源使用和信息流进行建模。标准操作语义无法强制执行对资源使用和信息流的约束。不过，基于环境的语义（例如堆语义）可以做到这一点。因此，本节将介绍一种基于加权堆的 LDC 语义。我们将使用同一种堆语义来分析 LDC 中的资源使用和信息流，正如我们在两种分析中使用同一种标准小步语义一样。

文献中已使用基于堆的语义来分析资源使用 [Choudhury 等人 2021；Marshall 等人 2022；Turner 和 Wadler 1999]。我们的堆语义遵循 Choudhury 等人 [2021]。我们的堆语义与 Choudhury 等人 [2021]的区别在于，我们将其同时用于用法分析和流分析，而他们仅将其用于用法分析。除此之外，两种语义本质上相同。

堆语义展示了一个项如何在为该项的自由变量赋予值的堆中进行归约。堆是由变量-项对组成的有序列表，其中的项可以被视为相应变量的定义。对于堆中的每个变量-项对，我们赋予一个权重，该权重可以是a $q \in \mathrm{Q}_{\mathbb{N}}$ 或an $\ell \in$ L。一个堆中每个变量-项对都具有关联权重时，称其为加权堆。我们假设我们的加权堆满足以下两个性质：唯一性，即一个变量不会被定义两次；以及无环性，即一个变量的定义不会引用自身，也不会引用堆中随后出现的其他变量。接下来，我们将归约建模为项与定义项的自由变量的加权堆之间的交互。

4.1 归约关系

基于堆的归约规则见图 7。关于这一归约关系 [H]a $\Longrightarrow^{q}_{S}$ [H′]a′，需要注意以下几点：

- S 在此表示一个支持集，[Pitts 2013]即在选择新名称时必须避开的变量集合。
- 从资源使用的角度来看，上述判断可以解读为：将 a 的 q 个副本所使用的资源从堆 H 中取出，生成 a′ 的 q 个副本，而 H′ 则是剩余资源。若将堆视为内存，那么资源的使用就对应于归约过程中的内存查找。

$$\text{Heap}, H ::= \emptyset \mid H, x \overset{q}{\mapsto} a$$

$[H]a \Longrightarrow_S^q [H']a'$ （选定的堆步进规则）

堆步进-变量

$$\frac{q \neq 0}{[H_1, x \overset{r+q}{\mapsto} a, H_2]x \Longrightarrow_S^q [H_1, x \overset{r}{\mapsto} a, H_2]a}$$

HeapStep-Discard

$$\frac{[H]a \Longrightarrow_S^q [H']a' \qquad q <: q'}{[H]a \Longrightarrow_S^{q'} [H']a'}$$

堆步进-左应用

$$\frac{[H]b \Longrightarrow_S^q [H']b'}{[H]b\ a^r \Longrightarrow_S^q [H']b'\ a^r}$$

HeapStep-AppBeta

$$\frac{y \text{ 新鲜} \qquad b' = b\{y/x\}}{[H](\lambda^r x{:}\ A.b)\ a^r \Longrightarrow_S^q [H, y \overset{q \cdot r}{\mapsto} a]\ b'}$$

堆步进-配对绑定β

$$\frac{x' \text{ 新鲜} \qquad y' \text{ 新鲜} \qquad b' = b\{x'/x\}\{y'/y\}}{[H] \text{令} q_0 (x^r, y) \text{是} (a_1{}^r, a_2) \text{于} b \Longrightarrow_S^q [H, x' \overset{q \cdot q_0 \cdot r}{\mapsto} a_1, y' \overset{q \cdot q_0}{\mapsto} a_2]b'}$$

堆步进-情形1 Beta

$$\frac{x_1' \text{ 新鲜} \qquad b_1' = b_1\{x_1'/x_1\}}{[H]\text{case 表达式}_{q_0}\ (\text{inj}_1\ a_1) \text{ of } x_1.b_1 ; x_2.b_2 \Longrightarrow_S^q [H, x_1' \overset{q \cdot q_0}{\mapsto} a_1]\ b_1'}$$

图 7. LDC 的堆语义（节选）

- 从信息流的角度来看，该判断可以解读为：在 H 的安全约束下，项a 转换为a′，其安全级别为 q，其中 H′ 是更新后的安全约束。将堆视为内存时，赋值上的安全标签对应于数据的访问权限，而判断上的标签对应于用户的安全许可。

现在，我们来考察图 7 中给出的一些归约规则。其中最有意思的是规则 HeapStep-Var。从资源使用的角度来看，这条规则可以理解为：对 x 执行 q 次步进，就需要查找 x 的值 q 次，因此会消耗 q 个资源。从信息流的角度来看，这条规则可以理解为：x 所指向、处于安全级别 $r \sqcap q$ 的数据，对 q 可观察，因为 $r \sqcap q \sqsubseteq q$。请注意，由于 $r \sqcap q \sqsubseteq r$，更新后的堆中赋值的安全级别不会降低。不过，它始终可以保持不变，因为 $(r \sqcap q) \sqcap q = r \sqcap q$。规则 HeapStep-Var 包含前置条件 $q \neq 0$，因为在世界 0 或 $\top$ 中，资源使用和流约束没有意义。

规则 HeapStep-Discard 也很有意思。从资源使用的角度来看，它允许在许可的情况下丢弃资源。从信息流的角度来看，它对应于随着观察者的安全许可级别提高，信息仍对观察者可见。

规则 堆步进-左应用 展示了 $b\ a^r$ 如何步进，因为 b 也会步进。请注意，前提中的支撑集包含参数的自由变量，以确保在选择新鲜名称时避开这些变量。我们省略了与其他消去形式对应的左规则，这些规则符合预期。

在贝塔规则 HeapStep-AppBeta、 HeapStep-LetPairBeta和 堆步进-情形1Beta中，我们向堆中添加新的赋值，同时避免变量捕获。添加新赋值时所使用的权重，由项上的注释和判断上的标签决定。例如，在

规则 HeapStep-AppBeta，新假设被添加到 $q \cdot r$ 或 $q \sqcup r$，因为应用上的注释是 r，且判断上的标签是 q。

根据这些规则，我们来看看一些能够进行的归约，以及一些不能 t.

$$[x \overset{1}{\mapsto} \text{true}]x \Longrightarrow_S^1 [x \overset{0}{\mapsto} \text{true}]\text{true but not } [x \overset{0}{\mapsto} \text{true}]x \Longrightarrow_S^1 [x \overset{0}{\mapsto} \text{true}]\text{true}$$

$$[x \overset{2}{\mapsto} \text{true}]x \Longrightarrow_S^2 [x \overset{0}{\mapsto} \text{true}]\text{true but not } [x \overset{1}{\mapsto} \text{true}]x \Longrightarrow_S^2 [x \overset{0}{\mapsto} \text{true}]\text{true}$$

$$[x \overset{\mathrm{L}}{\mapsto} \text{true}]x \Longrightarrow_S^{\mathrm{H}} [x \overset{\mathrm{L}}{\mapsto} \text{true}]\text{true but not } [x \overset{\mathrm{H}}{\mapsto} \text{true}]x \Longrightarrow_S^{\mathrm{L}} [x \overset{\mathrm{H}}{\mapsto} \text{true}]\text{true}$$

4.2 确保资源使用公平且信息流安全

观察图 7 中的规则，我们发现它们确保了资源使用的公平性。唯一允许使用资源的规则是堆步进-变量 规则。该规则确保只有在环境能够提供足够资源时，查找操作才会进行。成功查找后，它还会从环境中取走必要的资源。图 7 中的规则也确保了信息流的安全性。唯一允许信息从堆流向程序的规则同样是 堆步进-变量 规则。该规则确保只有在用户拥有必要权限时，信息才能流动。

以下两个引理将上述论证形式化。第一个引理指出，在归约过程中的某个时刻不可用的定义，在之后也不会变得可用。第二个引理指出，不可用的定义不会对归约产生任何作用。请注意，在信息流的情形下，约束 $\neg(\exists q_0, r = q + q_0)$ 等价于 $\neg(\mathrm{r} \sqsubseteq \mathrm{q})$。（这里，$|\mathrm{H}|$ 表示 H 的长度。）

引理 4.1（不变）。如果 $[\mathrm{H}_1, \mathrm{x} \overset{r}{\mapsto} \mathrm{a}, \mathrm{H}_2]\mathrm{c} \Longrightarrow_S^q [\mathrm{H}\prime_1, \mathrm{x} \overset{r'}{\mapsto} \mathrm{a}, \mathrm{H}\prime_2]\mathrm{c}'$（其中 $|H_1| = |H_1'|$）且 $\neg(\exists q_0, r = q + q_0)$，那么 $\mathrm{r}\prime = r$。

引理 4.2（无关）。如果 $[\mathrm{H}_1, \mathrm{x} \overset{r}{\mapsto} \mathrm{a}, \mathrm{H}_2]\mathrm{c} \Longrightarrow_{S \cup \mathrm{fv}_b}^q [\mathrm{H}\prime 1, \mathrm{x} \overset{r'}{\mapsto} \mathrm{a}, \mathrm{H}\prime_2]\mathrm{c}'$（其中 $|\mathrm{H}_1| = |\mathrm{H}\prime_1|$）且 $\neg(\exists q_0, r = q + q_0)$，那么 $[\mathrm{H}_1, \mathrm{x} \overset{r}{\mapsto} \mathrm{b}, \mathrm{H}_2]\mathrm{c} \Longrightarrow_{S \cup \mathrm{fv}_a}^q [\mathrm{H}\prime 1, \mathrm{x} \overset{r'}{\mapsto} \mathrm{b}, \mathrm{H}\prime_2]\mathrm{c}'$。

这些引理与我们接下来给出的可靠性定理相结合，将确保 LDC 中资源使用的公平性和信息流安全性。

4.3 堆语义下的可靠性

基于堆语义进行用法分析的核心思想是：如果一个堆包含足够的资源，能够按照类型系统的判定来求值某个项的若干个副本，那么这些副本在该堆中的求值就不会陷入停滞。由于基于堆的归约规则保证了资源使用的公平性，这意味着类型系统能够正确地核算资源使用。

基于堆语义进行依赖分析的核心思想与此类似。如果一个堆按照类型系统的判定为某个用户设置了正确的访问权限，那么该堆中任何对该用户可观察的程序都不会在求值时陷入停滞。由于归约规则保证了信息流安全性，这意味着类型系统只允许安全的信息流。

兼容性关系 $H \models \Gamma$，用于描述堆 H 与上下文 Γ 之间的关系，它形式化地表达了这样的思想：堆 H 包含了正确数量的资源，或设置了正确的访问权限，以便对在上下文 Γ 中经过类型检查的任意项进行求值。兼容性关系 [由 Choudhury 等人 2021]定义如下：

堆 $\models \Gamma$ *(兼容性)*

堆兼容-空

$$\frac{}{\emptyset \models \emptyset}$$

HeapCompat-Cons

$$\frac{H \models \Gamma_1 + \Gamma_2 \quad \Gamma_2 \vdash a :^q A}{H, x \overset{q}{\mapsto} a \models \Gamma_1, x :^q A}$$

接下来陈述的可靠性定理指出，如果堆 H 与上下文 Γ 兼容，那么从堆 H 开始对在上下文 Γ 中经过类型检查的项求值时，不会陷入停滞。

定理 4.3（可靠性）。如果 堆 $\models \Gamma$ 且 $\Gamma \vdash a :^q A$ 且 $q \neq 0$，那么 a 要么是一个值，要么存在堆，Γ', a'，使得：

- $[\mathrm{H}]a \Longrightarrow_S^q [\mathrm{H}',]a'$
- $H' \models \Gamma'$
- $\Gamma' \vdash a' :^q A$

请注意，与标准的类型保持性定理不同，类型上下文会随着每一步归约而更新。这是因为随着项不断归约，它从上下文中所需的资源会减少。不过，更新后的上下文始终与更新后的堆兼容。此外，请注意，类型判断和步进判断上的标签是相同的。从资源使用的角度看，这种对应关系意味着归约过程中项的副本数量保持不变。从信息流的角度看，这种对应关系意味着归约过程中观察者级别保持不变。

下面给出该可靠性定理的一些推论。

推论 4.4（无使用）。在 *LDC($\mathcal{Q}_{\mathbb{N}}$)* 中：令 $\emptyset \vdash f :^{1\ 0} A \to A$。那么，对于任意 $\emptyset \vdash a_1 :^0 A$ 和 $\emptyset \vdash a_2 :^0 A$，项 $f\,a_1{}^0$ 和 $f\,a_2{}^0$ 具有相同的操作行为，即这两个项要么都发散，要么都归约为相同的值。

上述推论在 LDC(L) 中同样成立，只需分别将 0 和 1 替换为 H 和 L。在 LDC(L) 中，该推论表明高安全级别输入对低安全级别输出的非干扰。

推论 4.5（非干扰）。在 *LDC($\mathcal{L}$)* 中：设 $\emptyset \vdash f :^{\mathrm{L}\ \mathrm{H}} A \to A$。那么，对于任意 $\emptyset \vdash a_1 :^{\mathrm{H}} A$ 和 $\emptyset \vdash a_2 :^{\mathrm{H}} A$，项 $f\,a_1{}^{\mathrm{H}}$ 和 $f\,a_2{}^{\mathrm{H}}$ 具有相同的操作行为。

可靠性定理也可用于推理其他使用方式，例如 LDC($\mathbb{N}_{\geq}$) 中的仿射使用。

推论 4.6（仿射使用）。在 *LDC($\mathbb{N}_{\geq}$)* 中：设 $\emptyset \vdash f :^{1\ 1} A \to A$。那么，对于任意 $\emptyset \vdash a :^1 A$，项 $f\,a^1$ 在归约过程中最多使用 a 一次。

既然我们已经了解了 LDC 简单类型版本的语法和语义，接下来将转向其纯类型系统（PTS）版本。

5 纯类型系统中的线性与依赖分析

纯类型系统（PTS）由一个元组 $(\mathcal{S},\mathcal{A},\mathcal{R})$刻画，其中 $\mathcal{S}$ 是种类的集合，$\mathcal{A}$ 是公理的集合，而 $\mathcal{R}$ 是种类之间的三元关系 [Barendregt 1993]。许多类型系统，如简单类型的 λ演算、System F、SystemFω、构造演算、Type-in-Type 等，都可以视为 PTS。请注意，PTS不一定具有归一化性质；例如，Type-in-Type 允许非终止的计算。我们将LDC 参数化为一个抽象的PTS，以便根据需要将其实例化为特定的 PTS。

5.1 简单类型版本与 PTS 版本

LDC的 PTS版本与其简单类型版本相似。就类型和项而言，我们只需将 S中的种类添加到其中，并分别将 $^r\mathsf{A} \to \mathsf{B}$ 和 $^r\mathsf{A} \times \mathsf{B}$ 推广为 $\Pi \mathsf{x} :^r \mathsf{A}.\mathsf{B}$ 和 $\Sigma \mathsf{x} :^r \mathsf{A}.\mathsf{B}$。对于资源使用和信息流分析，我们使用相同的参数化结构，即 $\mathsf{Q}_{\mathbb{N}}$ 和 L。然而，这两个版本之间存在一个重要区别。在 PTS 版本中，我们需要将分析对象从项扩展到类型和项。

我们扩展背后的关键思想是，可以分别对类型和项进行使用分析与流分析。这一思想受到近期分级上下文依赖类型系统发展的启发 [Atkey 2018； Choudhury 等人 2021； McBride 2016； Moon 等人 2021]。 McBride [2016] 首先指出，通过区分类型中对资源的 “ 思考性使用 ” 和项中对资源的“消耗性使用”，可以将线性性与依赖类型平滑地结合起来。 Atkey [2018]通过演算 QTT推进了这项工作，其中类型存在于资源无关世界，而项存在于资源感知世界。 Choudhury 等人 [2021] 提出了另一种系统：GRAD，其中类型和项都存在于资源感知世界，但在计算该类型的项所使用的资源时，会将类型使用的资源归零。 Moon 等人 [2021]又提出了另一种系统， GRTT，其中类型使用的资源不会被归零，而是与项使用的资源同时进行跟踪。在其分析中，GRAD 比QTT更加统一，也比 GRTT简单得多。因此，LDC 按照 GRAD的方式分析类型中的使用情况。

我们现在考察该演算的类型系统。

5.2 LDC 的类型系统

类型判断和等式规则分别见图8 和9 。有几点需要注意：

- 我们将这些规则同时用于资源使用分析和信息流分析（并对 $+, \cdot, 0, 1$ 和 $<:$ 作适当解释）。
- 判断 $\Delta \vdash_0 \mathsf{a} : \mathsf{A}$ 是判断 $\Gamma \vdash \mathsf{a} :^0 \mathsf{A}$ 的简写，其中 $\lfloor \Gamma \rfloor = \Delta$ 和 Γ 是一个 0 向量。请注意，这个判断本质上等同于标准类型判断 $\Delta \vdash \mathsf{a} : \mathsf{A}$，因为在世界 0 中，资源使用和信息流流约束都没有意义。
- 我们分别根据项和类型跟踪使用情况与流动情况。规则 PTS-VAR 很好地体现了这一原则。类型 A 可能会使用一些资源，或在某个较低安全级别上可观察到。但在对类型为 A 的项进行类型检查时，我们会将 A 的要求归零，或将 A 设置为最高安全级别。这一原则也适用于其他几条规则，例如规则 PTS-LAM、规则 PTS-PAIR 等。
- 规则 PTS-PI展示了我们如何在类型中追踪用法和流动。该规则突显了分析的一个重要方面：我们不仅将类型和项中的分析分开，还允许项及其类型以不同方式处理同一个受约束变量。类型上的注释 r（在本例中）表示该类型所对应的项的主体中如何使用受约束变量。此注释与类型自身主体中受约束变量的用法无关。来看一个例子：多态恒等类型 $\Pi x :^0 s.\Pi y :^1 x.x$ 在其主体中使用了受约束变量 x，但具有此类型的函数（例如，多态恒等函数$\lambda^0 x.\lambda^1 y.y$）不能在其主体中使用受约束变量 x。
- 我们使用 β-等价性来表示规则 PTS-Conv规则。它是一个同余的、在 β-归约下对项封闭的等价关系。部分等式规则出现在图 9 中。它们大多是标准的。然而，同余规则 EQ-PICONG、EQ-LAMCONG和 EQ-APP-CONG 需要检查被置于等式两端的项上的等级标注是否匹配。

$\boxed{\Gamma \vdash a :^{q} A}$ （*PTS* 版本）

PTS-公理
$$\frac{c:\ s \in A}{\emptyset \vdash c :^{q} s}$$

PTS-Var
$$\frac{\Delta \vdash_0 A : s \qquad \lfloor \Gamma \rfloor = \Delta}{0 \cdot \Gamma, x :^{q} A \vdash x :^{q} A}$$

PTS-Weak 规则
$$\frac{\Gamma \vdash a :^{q} A \qquad \Delta \vdash_0 B : s \qquad \lfloor \Gamma \rfloor = \Delta}{\Gamma,\ y :^{0} B \vdash a :^{q} A}$$

PTS-Pi
$$\frac{\Gamma_1 \vdash A :^{q} s_1 \qquad \Gamma_2,\ x :^{r_0} A \vdash B :^{q} s_2 \qquad R(s_1, s_2, s_3 s) \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash \Pi x :^{r} A.B :^{q} s_3}$$

PTS-Lam
$$\frac{\Gamma, x :^{q \cdot r} A \vdash b :^{q} B \qquad \Delta \vdash_0 \Pi x :^{r} A.B : s \qquad \lfloor \Gamma \rfloor = \Delta}{\Gamma \vdash \lambda^{r} x{:}A.b :^{q} \Pi x :^{r} A.B}$$

PTS-应用
$$\frac{\Gamma_1 \vdash b :^{q} \Pi x :^{r} A.B \qquad \Gamma_2 \vdash a :^{q \cdot r} A \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash b\ a^{r} :^{q} B\{a/x\}}$$

PTS-Conv 规则
$$\frac{\Gamma \vdash a :^{q} A \qquad \Delta \vdash_0 B : s \qquad A =_{\beta} B \qquad \lfloor \Gamma \rfloor = \Delta}{\Gamma \vdash a :^{q} B}$$

PTS-Sigma
$$\frac{\Gamma_1 \vdash A_1 :^{q} s_1 \qquad \Gamma_2, x :^{r_0} A_1 \vdash A_2 :^{q} s_2 \qquad R(s_1, s_2, s_3) \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash \Sigma x :^{r} A_1.A_2 :^{q} s_3}$$

PTS-Pair
$$\frac{\Delta \vdash_0 \Sigma x :^{r} A_1.A_2 : s \qquad \Gamma_1 \vdash a_1 :^{q \cdot r} A_1 \qquad \Gamma_2 \vdash a_2 :^{q} A_2\{a_1/x\} \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor = \Delta}{\Gamma_1 + \Gamma_2 \vdash (a_1{}^{r}, a_2) :^{q} \Sigma x :^{r} A_1.A_2}$$

PTS-LetPair
$$\frac{\Delta, z :\ \Sigma x :^{r} A_1.A_2 \vdash_0 B : s \qquad \Gamma_1 \vdash a :^{q \cdot q_0} \Sigma x :^{r} A_1.A_2 \qquad \Gamma_2, x :^{q \cdot q_0 \cdot r} A_1, y :^{q \cdot q_0} A_2 \vdash b :^{q} B\{(x^{r}, y)/z\} \qquad q_0 <: 1 \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor = \Delta}{\Gamma_1 + \Gamma_2 \vdash \mathrm{let}_{q_0} (x^{r},\ y)\ \mathrm{be\ a\ in\ b} :^{q} B\{a/z\}}$$

PTS-Sum
$$\frac{\Gamma_1 \vdash A_1 :^{q} s \qquad \Gamma_2 \vdash A_2 :^{q} \text{秒} \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash A_1 + A_2 :^{q} s}$$

PTS-Inj1
$$\frac{\Delta \vdash_0 A_1 + A_2 :\ s \qquad \Gamma \vdash a_1 :^{q} A_1 \qquad \lfloor \Gamma \rfloor = \Delta}{\Gamma \vdash \mathrm{inj}_1\ a_1 :^{q} A_1 + A_2}$$

PTS-Case
$$\frac{\Delta,\ z : A_1 + A_2 \vdash_0 B : s \qquad \Gamma_1 \vdash a :^{q \cdot q_0} A_1 + A_2 \qquad \Gamma_2, x_1 :^{q \cdot q_0} A_1 \vdash b_1 :^{q} B\{\mathrm{inj}_1\ x_1/z\} \qquad \Gamma_2, x_2 :^{q \cdot q_0} A_2 \vdash b_2 :^{q} B\{\mathrm{inj}_2\ x_2/z\} \qquad q_0 <: 1 \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor = \Delta}{\Gamma_1 + \Gamma_2 \vdash \text{case 表达式}\ a\ \text{的}\ x_1.b_1 ; x_2.b_2 :^{q} B\{a/z\}}$$

PTS-SubL
$$\frac{\Gamma' \vdash a :^{q} A \qquad \Gamma <: \Gamma'}{\Gamma \vdash a :^{q} A}$$

PTS-SubR
$$\frac{\Gamma \vdash a :^{q} A \qquad q <: q'}{\Gamma \vdash a :^{q'} A}$$

图 8. LDC 的类型系统（摘录）

$\boxed{A =_\beta B}$ （定义相等）

$$\text{Eq-PiCong}\ \frac{A_1 =_\beta A_2 \qquad B_1 =_\beta B_2}{\Pi x :^r A_1.B_1 =_\beta \Pi x :^r A_2.B_2} \qquad \text{Eq-LamCong}\ \frac{A_1 =_\beta A_2 \qquad b_1 =_\beta b_2}{\lambda^r x : A_1.b_1 =_\beta \lambda^r x : A_2.b_2} \qquad \text{Eq-AppCong}\ \frac{b_1 =_\beta b_2 \qquad a_1 =_\beta a_2}{b_1 a_1{}^r =_\beta b_2\ a_2{}^r}$$

图 9. LDC 的等式规则（摘录）

接下来，我们来看该演算的元理论。

5.3 LDC 的元理论

LDC 的 PTS 版本满足简单类型版本所满足的所有引理和定理的 PTS 类比形式，这些内容在第 3.2 节和 3.4 节中给出。PTS 版本也采用与简单类型版本相同的标准按名调用语义。此外，PTS 版本相对于该语义是类型健全的。

接下来，我们陈述第 3.2 节和 3.4 节中所述若干关键引理和定理的 PTS 对应版本。

引理 5.1（替换）。若 $\Gamma_1, z :^{r_0} C\ \Gamma_2 \vdash a :^q A$ 且 $\Gamma \vdash c :^{r_0} C$ 且 $\lfloor\Gamma_1\rfloor = \lfloor\Gamma\rfloor$，则 $\Gamma_1 + \Gamma,\ \Gamma_2\{c/z\} \vdash a\{c/z\} :^q A\{c/z\}$。

定理 5.2（保持性）。若 $\Gamma \vdash a :^q A$ 且 $\vdash a \rightsquigarrow a'$，则 $\Gamma \vdash a' :^q A$。

定理 5.3（进展）。若 $\emptyset \vdash a :^q A$，则 a 是一个值，或者存在 a' 使得 $\vdash a \rightsquigarrow a'$。

接下来，我们考虑 LDC 的堆语义。

5.4 LDC 的堆语义

LDC 的 PTS 版本具有与其简单类型对应版本相同的堆归约关系。然而，PTS 版本在归约过程中项的良类型性方面面临挑战。在简单类型版本中，我们可以将项中的替换延迟到加载进堆中，而无需担心这会如何影响该项的类型。在 PTS 版本中，延迟替换可能导致项“落后于”类型。我们考虑一个来自 Choudhury 等人 [2021]的示例来说明这一点。

多态恒等函数 $\lambda^0 x : s.\lambda^1 y : x.y$ 的类型为 $\Pi x :^0 s.\Pi y :^1 x.x$。将该函数实例化为 Unit，可得 $(\lambda^0 x : s.\lambda^1 y : x.y)\ \text{Unit}^0$，其类型为 $\Pi y :^1 \text{Unit}.\text{Unit}$。现在，$[\emptyset](\lambda^0 x : s.\lambda^1 y : x.y)\ \text{Unit}^0 \Longrightarrow^1_S [x \overset{0}{\mapsto} \text{Unit}]\lambda^1 y : x.y$。除非查看堆中的定义，否则我们没有理由相信 $\lambda^1 y : x.y$ 的类型是 $\Pi y :^1 \text{Unit}.\text{Unit}$。延迟替换 $x \overset{0}{\mapsto} \text{Unit}$ 会导致项 $\lambda^1 y : x.y$ 落后于类型 $\Pi y :^1 \text{Unit}.\text{Unit}$。请注意，这一挑战仅由延迟替换引起，与使用分析和流分析无关。

为克服这一挑战，Choudhury 等人 [2021]采用了以下策略。首先，他们扩展类型系统，加入允许定义的上下文。然后，他们证明扩展后的演算相对于堆语义是可靠的。随后，他们证明原始演算与扩展后的演算等价。基于这一等价性，他们得出结论：原始演算相对于堆语义也是可靠的。我们对 LDC 采用了相同的策略。由于篇幅所限，本文正文省略了详细内容。感兴趣的读者请参阅附录F。

LDC 相对于堆语义是可靠的。请注意，下面的陈述与其简单类型版本相同。

定理5.4（可靠性）。 若$H \models \Gamma$ 且 $\Gamma \vdash a :^q A$ 以及 $q \neq 0$，则要么a 是一个值，要么存在 H，Γ'、a'，使得 $[H]a \Longrightarrow_S^q [H,]a'$ 且 H，$\models \Gamma'$ 且 $\Gamma' \vdash a' :^q A$。

第 4.3节中介绍的可靠性定理的推论同样适用于 PTS 版本。下面我们介绍一些与多态类型相关的推论。

推论 5.5. 在 *LDC(*$\mathcal{Q}_{\mathbb{N}}$*)* 中：如果 $\emptyset \vdash f :^1 \Pi x :^0 s.x$ 且 $\emptyset \vdash_0 A : s$，那么 $f\,A^0$ 必然发散。

推论 *5.6*。 在 *LDC(*$\mathcal{Q}_{\mathbb{N}}$*)* 中：在强规范化 *PTS* 中，如果 $\emptyset \vdash f :^1 \Pi x :^0 s.\Pi y :^1 x.x$ 且 $\emptyset \vdash_0 A :^1 s$ 且 $\emptyset \vdash a :^1 A$，则 $f\,A^0\,a^1 =_\beta a$。

6 添加不受限使用

到目前为止，我们一直使用 LDC 分析精确用量、有界用法和依赖关系。本节中，我们也使用演算来分析无限制使用。由 ω表示的无限制使用与由 $n \in \mathbb{N}$表示的精确用量和有界用法有两点不同：

- ω 是加法湮灭元，意味着对所有 $q \in \mathbb{N} \cup \{\omega\}$，$\omega + q = q + \omega = \omega$。没有 $n \in \mathbb{N}$ 是加法湮灭元。
- ω 也几乎是乘法零化子，这意味着对于 $q \in (\mathbb{N} - \{0\}) \cup \{\omega\}$，$\omega \cdot q = q \cdot \omega = \omega$。没有 $n \in \mathbb{N} - \{0\}$是乘法零化子。

为了适应 ω 的这种行为，我们需要对类型系统进行更改。但在进行这项更改之前，让我们先修正我们的预序半环：

- $\mathbb{N}^{\omega}_{=}$，其中包含 ω 和预序半环 $\mathbb{N}_{=}$，且 $\omega <:$ 对所有 q
- $\mathbb{N}^{\omega}_{\geq}$，其中包含 ω 和预序半环 $\mathbb{N}_{\geq}$，且 $\omega <:$ 对所有 q
- $\mathcal{Q}$线性与$\mathcal{Q}$仿射，见第2.2节

我们用 $\mathrm{Q}^{\omega}_{\mathbb{N}}$ 表示上述半环集合中的任意一个元素。接下来，我们讨论从 LDC($\mathrm{Q}_{\mathbb{N}}$) 转到 LDC($\mathrm{Q}^{\omega}_{\mathbb{N}}$) 时所需的变更。

6.1 问题及其解决方案

允许无限制使用时，图4 和8 中的类型系统无法保证资源使用的公平性。请看以下由简单类型系统允许的“不公平”推导：

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x :^{\omega} A \vdash x :^{\omega} A}\ \text{ST-Var} \quad \dfrac{}{x :^{\omega} A \vdash x :^{\omega} A}\ \text{ST-Var}}{x :^{\omega} A \vdash (x^1, x) :^{\omega} {}^1A \times A}\ \text{ST-Pair}}{\emptyset \vdash \lambda^1 x : A.(x^1, x) :^{\omega} {}^1A \to {}^1A \times A}\ \text{ST-Lam}}{\emptyset \vdash \lambda^1 x : A.(x^1, x) :^1 {}^1A \to {}^1A \times A}\ \text{ST-SubR}$$

判断 $\emptyset \vdash \lambda^1$x: A.(x^1, x) :^{1} ^{1}A $\to$ ^{1}A $\times$A不公平，因为它允许复制资源。仔细观察推导，我们发现，当 ω “欺骗”ST-Lam 规则，使其相信该项只使用 x 一次时，不公平性就产生了。

这种不公平性导致了类型可靠性的失败。要了解其中的原因，请考虑以下项：y :^{1} A $\vdash$ (λ^1x: A.(x^1, x)) y^1 :^{1} ^{1}A $\times$A，它通过上述推导和规则 ST-App进行了类型检查。该项规约为：$\vdash$ (λ^1x : A.(x^1, x)) y^1 $\rightsquigarrow$ (y^1, y)。但随后会出现不可靠性，因为：y :^{1} A0 (y^1,y) :^{1} ^{1}A $\times$ A。因此，为了确保类型可靠性，我们需要修改规则 ST-Lam 和规则 PTS-Lam。

我们对这些规则作如下修改：

$$\textsc{ST-LamOmega}\ \frac{\Gamma, x :^{q\cdot r} A \vdash b :^{q} B \qquad q = \omega \Rightarrow r = \omega \qquad q_0 \neq 0}{q_0 \cdot \Gamma \vdash \lambda^{r} x : A.b :^{q_0 \cdot q\ r} A \to B} \qquad \textsc{PTS-LamOmega}\ \frac{\Gamma, x :^{q\cdot r} A \vdash b :^{q} B \qquad \Delta \vdash_0 \Pi x :^{r} A.B : s \qquad \lfloor \Gamma \rfloor = \Delta \qquad q = \omega \Rightarrow r = \omega \qquad q_0 \neq 0}{q_0 \cdot \Gamma \vdash \lambda^{r} x : A.b :^{q_0 \cdot q}\ \Pi x :^{r} A.B}$$

关于上述规则，有几点需要注意：

- 规则ST-LamOmega 和PTS-LamOmega 是规则 ST-Lam 和PTS-Lam的推广形式，也就是说，当等级限制为自然数时，用规则 ST-Lam和PTS-Lam 替换为规则 ST-LamOmega 和 PTS-LamOmega 对类型系统没有影响。
- 这些规则施加了约束：$\mathsf{q} = \omega \Rightarrow \mathsf{r} = \omega$。这样一来 ω 就无法“欺骗” Lambda 规则，使其误以为函数对参数的使用程度低于实际情况。尤其是，采用这些修改后的规则后，上述不公平推导将无法成立。
- 约束 $\mathsf{q} = \omega \Rightarrow \mathsf{r} = \omega$ 虽然是阻止不公平推导所必需的，但也会阻止一些公平推导，例如下面这个：

$$\frac{x :^{\omega} A \vdash x :^{\omega} A}{\emptyset \vdash \lambda^{1} x : A.x :^{\omega\ 1} A \to A}$$

  为了允许此类推导，同时仍施加这一约束，规则 ST-LamOmega 和 PT-S-LamOmega 将结论判断乘以 q_0。这种乘法使这些规则能够允许上述推导：

$$\frac{x :^{1} A \vdash x :^{1} A}{\emptyset \vdash \lambda^{1} x : A.x :^{\omega\ 1} A \to A}$$

- 侧条件 $\mathsf{q}_0 \neq 0$ 可确保有意义的判断不会变成无意义的判断。请回想一下，就线性性和依赖关系分析而言，0 世界中的判断毫无意义。
- 规则 ST-LamOmega也可以等价地替换为以下两个更简单的规则（对于规则PTS-LamOmega同样如此）：

$$\textsc{ST-LamOmega0}\ \frac{\Gamma, x :^{q\cdot r} A \vdash b :^{q} B \qquad q = \omega \Rightarrow r = \omega}{\Gamma \vdash \lambda^{r} x : A.b :^{q\ r} A \to B} \qquad \textsc{ST-Omega}\ \frac{\omega \cdot \Gamma \vdash a :^{q} A \qquad q \neq 0}{\omega \cdot \Gamma \vdash a :^{\omega} A}$$

将规则 ST-Lam 和 PTS-Lam 替换为规则 ST-LamOmega 和 PTS-LamOmega，是我们为使这些类型系统能够追踪无限制使用而对图 4 和 8 所示类型系统所需进行的唯一修改。

通过对类型系统进行此修改，LDC($Q_{\mathbb{N}}^{\omega}$) 满足 LDC($Q_{\mathbb{N}}$) 所满足的所有引理和定理。特别是，LDC($Q_{\mathbb{N}}^{\omega}$) 满足类型可靠性（定理 5.2 和 5.3）以及堆可靠性（定理 5.4）。我们在下文中将此性质表述为一个定理。

定理 6.1。 *LDC($Q_{\mathbb{N}}^{\omega}$) 满足类型可靠性和堆可靠性。*

因此，LDC 是一种通用的线性依赖演算。跟踪线性性时，我们可以使用任意一个 $\mathsf{Q}_{\mathbb{N}}^{\omega}$s。跟踪依赖关系时，我们可以使用任意一个格。若要同时跟踪线性性和依赖关系，我们可以使用这些结构的笛卡尔积。下文将介绍一些示例，说明如何在 LDC 中结合线性性与依赖分析。

6.2 组合线性性与依赖分析

首先请注意，尽管预序半环和格的公理集合截然不同，无法相互协调（如第 2.3 节所述），但它们仍然是具有两个二元运算符和一个二元序关系的代数结构。因此，可以定义这两种结构的笛卡尔积。具体而言，给定一个预序半环 $\mathsf{Q} = (Q, +, \cdot\, 0, 1, <:)$ 和一个格 $\mathsf{L} = (L, \sqcap, \sqcup, \top, \bot, \sqsubseteq)$，可以将 $\mathsf{Q} \times \mathsf{L}$（即 Q 与 L 的笛卡尔积）定义为集合 $Q \times L$，并配以：

- 两个常量 0 和 1，定义为：0，$(0, \top)$ 和1，$(1, \bot)$；
- 一个二元运算符 $+$，定义为：$(q_1, \ell_1) + (q_2, \ell_2) \triangleq (q_1 + q_2, \ell_1 \sqcap \ell_2)$；
- 另一个二元运算符 $\cdot$，定义如下：$(q_1, \ell_1) \cdot (q_2, \ell_2) \triangleq (q_1 \cdot q_2, \ell_1 \sqcup \ell_2)$; 且
- 二元序关系，$<:$，定义为：$(q_1, \ell_1) <: (q_2, \ell_2) \triangleq q_1 <: q_2$ 且 $\ell_1 \sqsubseteq \ell_2$。

现在，给定任意 $\mathsf{Q}^{\omega}_{\mathbb{N}}$和 L，我们可以将 LDC 关于 $\mathsf{Q}^{\omega}_{\mathbb{N}} \times \mathsf{L}$ 进行参数化，利用上述对 $+, \cdot, 0, 1$和 $<:$ 的解释。这样的参数化有助于我们结合线性性和依赖性分析。举例来说，令 $\mathsf{Q} = \mathsf{Q}_{\mathrm{Lin}}$（或 $\mathsf{Q}_{\mathrm{Aff}}$）和 $\mathsf{L} = \mathsf{L} \sqsubseteq \mathsf{M} \sqsubseteq \mathsf{H}$。然后，在 LDC($\mathsf{Q} \times \mathsf{L}$)（扩展了 Int 类型）中，我们有：

$$x :^{(\omega,\mathrm{L})} \mathrm{Int} \vdash x + x :^{(1,\mathrm{L})} \mathrm{Int}$$

$$x :^{(\omega,\mathrm{L})} \mathrm{Int}, y :^{(1,\mathrm{M})} \mathrm{Bool} \vdash \mathrm{if}\ y\ \mathrm{then}\ x + x\ \mathrm{else}\ x :^{(1,\mathrm{M})} \mathrm{Int}.$$

注意，如果我们将任一判断中的假设$\mathsf{x}:^{(\omega,\mathrm{L})} \mathrm{Int}$ 改为$\mathsf{x} :^{(1,\mathrm{L})} \mathrm{Int}$ 或 $\mathsf{x}:^{(\omega,\mathrm{H})} \mathrm{Int}$，那么该判断将不再成立。类似地，如果我们将第二个判断中的假设$\mathsf{y}:^{(1,\mathrm{M})} \mathrm{Bool}$ 改为 $\mathsf{y} :^{(1,\mathrm{H})} \mathrm{Bool}$，那么它也将不再成立。

7 LDC 与标准线性和依赖关系演算系统

7.1 与简单类型演算系统的比较

在第 3.5 节中，我们讨论了标准依赖演算封装演算如何嵌入 LDC($\mathcal{L}$)。现在，我们将 LDC 与标准线性演算——线性非线性（LNL） λ-演算 Benton [1994] 进行比较。LNL λ-演算仅跟踪线性和无限制使用，并且是简单类型的。因此，我们将其与简单类型的 LDC(Q_{Lin}) 进行比较。下面，我们给出从 LNL λ-演算到 LDC(Q_{Lin}) 的保义翻译。

LNL演算采用两种形式的上下文和两种形式的类型判断。这两种形式的上下文，即线性上下文和非线性上下文，分别对应于LDC($\mathsf{Q}_{\mathrm{Lin}}$)中等级$1$ 和 ω 的假设。这两种形式的判断，即线性判断和非线性判断，分别对应于 LDC($\mathsf{Q}_{\mathrm{Lin}}$)中世界$1$和 ω 中的推导。在LNL λ演算中，线性上下文和非线性上下文分别记作 Γ 和 Θ，而线性判断和非线性判断分别写作 $\Theta;\ \Gamma \vdash_{\mathcal{L}} \mathsf{e}: \mathsf{A}$和 $\Theta \vdash_{C} \mathsf{t}:\mathsf{X}$。该演算包含标准直觉主义类型和线性类型；它还包含两个类型构造器 F和 G，线性世界和非线性世界通过它们进行交互。该演算使用A, B 表示线性类型；X, Y 表示非线性类型；a, b 表示线性变量；x, y 表示非线性变量；e, f 表示线性项；以及s, t 表示非线性项。

我们给出从LNL λ-演算 [Benton 1994]到 LDC(Q_{Lin}) 的图10中的翻译函数。该翻译保持类型判断和含义不变。

定理 7.1. 从 *LNL* λ-演算到 *LDC(Q_{Lin})* 的翻译（如图 *10*所示）是可靠的：

- 若 $\Theta;\ \Gamma \vdash_{\mathcal{L}} e : A$，则 $\overline{\Theta}^{\omega}, \overline{\Gamma}^{1} \vdash \overline{e} :^{1} \overline{A}$。
- 若 $\Theta \vdash_{C} t : X$，则 $\overline{\Theta}^{\omega} \vdash \overline{t} :^{\omega} \overline{X}$。
- 如果$e =_{\beta} f$，那么$\overline{e} =_{\beta} \overline{f}$。如果$s =_{\beta} t$，那么$\overline{s} =_{\beta} \overline{t}$。

$$\overline{1} = \mathrm{Unit} \qquad \overline{X \times Y} = {}^{\omega}\overline{X} \times \overline{Y} \qquad \overline{I} = \mathrm{Unit} \qquad \overline{A \otimes B} = {}^{1}\overline{A} \times \overline{B}$$

$$\overline{X \to Y} = {}^{\omega}\overline{X} \to \overline{Y} \qquad \overline{G\,A} = \overline{A} \qquad \overline{A \multimap B} = {}^{1}\overline{A} \to \overline{B} \qquad \overline{F\,X} = {}^{\omega}\overline{X} \times \mathrm{Unit}$$

$\overline{x} = x$	$\overline{a} = a$
$\overline{()} = \mathrm{unit}$	$\overline{*} = \mathrm{unit}$
$\overline{(s,t)} = (\overline{s}^{\omega}, \overline{t})$	$\overline{e \otimes f} = (\overline{e}^{1}, \overline{f})$
$\overline{\mathrm{fst}(s)} = \mathrm{let}_1\ (x^{\omega}, y)\ \mathrm{be}\ \overline{s}\ \mathrm{in}\ x\ \ (x, y\ \mathrm{fresh})$	$\overline{\mathrm{let}\ a \otimes b\ \mathrm{be}\ e\ \mathrm{in}\ f} = \mathrm{let}_1\ (a^{1}, b)\ \mathrm{be}\ \overline{e}\ \mathrm{in}\ \overline{f}$
$\overline{\mathrm{snd}(s)} = \mathrm{let}_1\ (x^{\omega}, y)\ \mathrm{be}\ \overline{s}\ \mathrm{in}\ y\ \ (x, y\ \mathrm{fresh})$	$\overline{\mathrm{let}\ *\ \mathrm{be}\ e\ \mathrm{in}\ f} = \mathrm{let}_1\ \mathrm{unit}\ \mathrm{be}\ \overline{e}\ \mathrm{in}\ \overline{f}$
$\overline{\lambda x : X.s} = \lambda^{\omega} x : \overline{X}.\overline{s}$	$\overline{\lambda a : A.e} = \lambda^{1} a : \overline{A}.\overline{e}$
$\overline{s\ t} = \overline{s}\ \overline{t}^{\omega}$	$\overline{e\ f} = \overline{e}\ \overline{f}^{1}$
$\overline{G\ e} = \overline{e}$	$\overline{\mathrm{derelict}\ s} = \overline{s}$
$\overline{F\ s} = (\overline{s}^{\omega}, \mathrm{unit})$	$\overline{\mathrm{let}\ Fx\ \mathrm{be}\ e\ \mathrm{in}\ f} = \mathrm{let}_1\ (x^{\omega}, y)\ \mathrm{be}\ \overline{e}\ \mathrm{in}\ \mathrm{let}_1\ \mathrm{unit}\ \mathrm{be}\ y\ \mathrm{in}\ \overline{f}$ $(y\ \mathrm{fresh})$

图 10. 从 LNL λ-演算系统到 LDC(QLin) 的类型和项翻译

这里，$\overline{\Gamma}^{1}$ 和 $\overline{\Theta}^{\omega}$ 分别表示 Γ 和 Θ，其中类型经过转换，假设的等级为1 和 ω。此外，$- =_{\beta} -$ 表示 LNLλ 演算 [Benton 1994]中各项之间的 β 等价关系。

LDC 的可靠性定理与上述保持含义的翻译相结合，表明 LDC($\mathrm{Q_{Lin}}$) 的表达能力不低于 LNL λ-演算系统。事实上，LDC($\mathrm{Q_{Lin}}$) 的表达能力强于 LNL λ-演算系统，因为后者没有对 0 使用进行建模。由于这一原因，从 LDC($\mathrm{Q_{Lin}}$) 到 LNL λ-演算系统的反向翻译无法实现！

7.2 与依赖类型演算系统的比较

接下来，我们将 LDC 与标准的依赖类型线性演算系统和依赖关系演算系统进行比较。正如第 5.1 节所述，依赖类型系统中存在若干用于线性性分析的演算系统。在这些演算系统中，LDC 与 GrAD （由 Choudhury 等人 [2021] 提出）最为接近。因此，我们将 LDC 与 GrAD 进行比较。

GraD 是一种通用的反效应演算，其参数化结构为任意偏序半环。另一方面，LDC($\mathrm{Q}_{\mathbb{N}}^{\omega}$) 是一种线性性演算，其参数化结构为特定的预序半环，即 $\mathrm{Q}_{\mathbb{N}}^{\omega}$s。比较这两种演算在这些半环上的表现，可以证明 LDC 包含 GraD。

定理 7.2。 以 $\boldsymbol{Q}_{\mathbb{N}}^{\omega}$ 作为参数化结构，若 $\Gamma \vdash a : A$ 在 *GraD* 中成立，则 $\Gamma \vdash a :^{1} A$ 在 *LDC* 中成立。此外，若 $\vdash a \rightsquigarrow a'$ 在 *GraD* 中成立，则 $\vdash a \rightsquigarrow a'$ 在 *LDC* 中成立。

T上述定理并不令人意外，因为我们在设计 LDC 时遵循了 GraD。

接下来，我们将 LDC 与标准的依赖演算系统进行比较。在相关文献中，只有少数几种演算系统 [Bernardy 和 Guilhem 2013；Choudhury 等人 2022；Prost 2000] 探讨了这一主题。其中，演算系统 $\mathrm{DDC}^{\top}$ [Choudhury 等人 2022]启发了 LDC 的设计，详见第 2.4 节。正因如此，就依赖分析而言，LDC 与 $\mathrm{DDC}^{\top}$相似。形式上，我们可以证明，当以任意格作为参数化结构时，LDC 包含 $\mathrm{DDC}^{\top}$。

定理 7.3。 以$\mathcal{L}$作为参数化结构，若 $\Gamma \vdash a :^{\ell} A$ 属于 $DDC^{\top}$，则 $\Gamma \vdash a :^{\ell} A$ 属于 LDC。此外，若 $\vdash a \rightsquigarrow a'$ 属于 $DDC^{\top}$，则 $\vdash a \rightsquigarrow a'$ 属于 LDC。

7.3 与其他相关演算系统的比较

在本文中，我们提出了关于用法分析和依赖分析的统一视角。这一论述有先例可循：Benton 和 Wadler [1996]指出，线性逻辑的模型也为Moggi的计算元语言提供模型。广义地说，用法分析可以被视为细粒度线性逻辑的实际运用，而依赖分析则可以被视为细粒度计算元语言的实际运用。因此，我们关于用法分析和依赖分析的统一视角，可以被视为对 Benton 和 Wadler [1996]所作观察的推广。

接下来，线性性和依赖关系传统上分别作为反效应和效应来分析。已有研究 [Gaboardi 等人 2016]探讨了如何结合效应与反效应。 Gaboardi 等人 [2016]提出了一种演算，它采用不同的分级模态来分析反效应和效应，并通过允许这些模态经由分级分配律相互作用，将二者结合起来。相比之下，LDC 使用相同的分级模态来分析线性性和依赖关系，但在两种分析中从不同的代数结构中取用等级。此外，LDC 使用的分级模态既是单子性的，也是余单子性的，而 Gaboardi 等人 [2016]使用的分级模态则是单子性的或余单子性的，但不一定二者兼具。这些差异背后的主要原因在于，Gaboardi 等人 [2016]的研究动机是构建一种用于结合反效应与效应的通用演算，而我们的动机则是构建一种用于结合线性性与依赖关系的专用演算。由于这种专用性，我们的演算不需要多种模态或分级分配律。最后请注意，Gaboardi 等人 [2016]提出的演算是简单类型的，而 LDC 支持依赖类型。

8 结论

我们已证明，线性性分析与依赖分析可以系统地统一并结合为一个演算系统。我们介绍了 LDC，这是一种用于纯类型系统中组合线性与依赖分析的通用演算。我们通过堆语义证明了 LDC 中的线性性分析与依赖分析是可靠的。我们还证明，LDC 涵盖了线性性分析与依赖分析的标准演算系统。本文重点讨论了 LDC 的句法性质。未来，我们计划探索该演算的语义性质。特别是，我们希望了解 LDC 的语义模型与线性类型系统和依赖类型系统的范畴模型有何异同。

参考文献

Martín Abadi、Anindya Banerjee、Nevin Heintze 和 Jon G. Riecke。1999。依赖核心演算。载于第 26 届 ACM SIGPLAN-SIGACT 编程语言原理研讨会论文集（美国得克萨斯州圣安东尼奥）（POPL '99）。计算机协会，美国纽约州纽约，147–160。https://doi.org/10.1145/292540.292555 Andreas Abel 和 Jean-Philippe Bernardy。2020。类型系统中模态的统一视角。ACM 编程语言会议录 4，ICFP，论文 90（2020 年 8 月），28 页。https://doi.org/10.1145/3408972 Samson Abramsky。1993。线性逻辑的计算解释。理论计算机科学 111，1（1993），3–57。https://doi.org/10.1016/0304-3975(93)90181-R Robert Atkey。2018。定量类型理论的语法与语义。载于第 33 届 ACM/IEEE 计算机科学逻辑年会论文集（英国牛津）（LICS '18）。计算机协会，美国纽约州纽约，56–65。https://doi.org/10.1145/3209108.3209189 Andrew Barber。1996。对偶直觉线性逻辑。技术报告。苏格兰爱丁堡。H. P. Barendregt。1993。带类型的 λ 演算。牛津大学出版社，美国，117–309。N. Benton 和 P. Wadler。1996。线性逻辑、单子与 λ 演算。载于第 11 届 IEEE 计算机科学逻辑年会论文集。420–431。https://doi.org/10.1109/LICS.1996.561458 P. N. Benton。1994。一种混合线性与非线性逻辑：证明、项与模型（扩展摘要）。载于第 8 届国际计算机科学逻辑研讨会精选论文（CSL '94）。施普林格出版社，柏林、海德堡，121–135。P. N. Benton、Gavin M. Bierman、Valeria de Paiva 和 Martin Hyland。1993。直觉线性逻辑项演算。载于类型化 λ 演算与应用国际会议论文集（TLCA '93）。施普林格出版社，柏林、海德堡，75–90。Jean-Philippe Bernardy、Mathieu Boespflug、Ryan R. Newton、Simon Peyton Jones 和 Arnaud Spiwack。2018。Linear Haskell：高阶多态语言中的实用线性性。载于 2018 年编程语言原理会议（POPL 2018）。Jean-Philippe Bernardy 和 Moulin Guilhem。2013。彩色类型理论。SIGPLAN 通讯 48，9（2013 年 9 月），61–72。https://doi.org/10.1145/2544174.2500577 G. Birkhoff。1967。格论（第 3 版）。美国数学学会，普罗维登斯。Edwin C. Brady。2021。Idris 2：实践中的定量类型理论。载于第 35 届欧洲面向对象编程会议 ECOOP 2021，2021 年 7 月 11–17 日，丹麦奥胡斯（线上会议）（LIPIcs，第 194 卷），Anders Møller 和 Manu Sridharan（编）。达格施图尔城堡—莱布尼茨信息学中心，9:1–9:26。https://doi.org/10.4230/LIPIcs.ECOOP.2021.9 Aloïs Brunel、Marco Gaboardi、Damiano Mazza 和 Steve Zdancewic。2014。定量共效应核心演算。载于第 23 届欧洲编程语言与系统研讨会论文集，第 8410 卷。施普林格出版社，柏林、海德堡，351–370。https://doi.org/10.1007/978-3-642-54833-8_19 Luís Caires、Frank Pfenning 和 Bernardo Toninho。2016。作为会话类型的线性逻辑命题。计算机科学中的数学结构 26，3（2016），367–423。https://doi.org/10.1017/S0960129514000218 Cristiano Calcagno、Walid Taha、Liwen Huang 和 Xavier Leroy。2003。使用抽象语法树、Gensym 和反射实现多阶段语言。载于生成式编程与组件工程，Frank Pfenning 和 Yannis Smaragdakis（编）。Springer Berlin Heidelberg，柏林、海德堡，57–76。Jawahar Chirimar、Carl Gunter 和 Jon Riecke。2000。引用计数作为线性逻辑的计算解释。函数式编程杂志 6（2000 年 3 月）。https://doi.org/10.1017/S0956796800001660 Pritam Choudhury。2022。依赖分析的单子与余单子方面。ACM 编程语言会议录 6，OOPSLA2，论文 172（2022 年 10 月），29 页。https://doi.org/10.1145/3563335 Pritam Choudhury、Harley Eades 和 Stephanie Weirich。2022。依赖型依赖演算。载于编程语言与系统，Ilya Sergey（编）。Springer International Publishing，Cham，403–430。Pritam Choudhury、Harley Eades III、Richard A. Eisenberg 和 Stephanie Weirich。2021。具有用量感知语义的分级依赖类型系统。ACM 编程语言会议录 5，POPL，论文 50（2021 年 1 月），32 页。https://doi.org/10.1145/3434331 Rowan Davies。2017。绑定时刻分析的时序逻辑方法。ACM 杂志 64，1，论文 1（2017 年 3 月），45 页。https://doi.org/10.1145/3011069 Dorothy E. Denning。1976。安全信息流的格模型。ACM 通讯 19，5（1976 年 5 月），236–243。https://doi.org/10.1145/360051.360056 Matthew Fluet、Greg Morrisett 和 Amal Ahmed。2006。线性区域就是你所需的一切。载于第 15 届欧洲编程语言与系统会议论文集（奥地利维也纳）（ESOP'06）。施普林格出版社，柏林、海德堡，7–21。https://doi.org/10.1007/11693024_2 Soichiro Fujii。2019。分级单子与索引单子的二范畴研究。arXiv:1904.08083 [math.CT] Marco Gaboardi、Shin-ya Katsumata、Dominic Orchard、Flavien Breuvart 和 Tarmo Uustalu。2016。通过分级结合效应与共效应。SIGPLAN 通讯 51，9（2016 年 9 月），476–489。https://doi.org/10.1145/3022670.2951939

Dan R. Ghica 和AlexI.Smith。2014。资源半环中的有界线性类型。载于《编程语言与系统》，Zhong Shao（编）。Springer Berlin Heidelberg，柏林，海德堡，331–350。Jean-Yves Girard。1987。线性逻辑。《理论计算机科学》50，1（1987），1–101。https://doi.org/10.1016/0304-3975(87)90045-4 Jean-YvesGirard、Andre Scedrov 和Philip J. Scott。1992。有界线性逻辑：一种多项式时间可计算性的模块化方法。《理论计算机科学》97，1（1992），1–66。https://doi.org/10.1016/0304-3975(92)90386-T CarstenK. Gomard 和Neil D.Jones。1991。无类型 λ 演算的部分求值器。《函数式编程杂志》1，1（1991），21–69。https://doi.org/10.1017/S0956796800000058 Nevin Heintze和JonG.Riecke。1998。SLam 演算：使用保密性与完整性进行编程。载于第25 届ACMSIGPLAN-SIGACT 编程语言原理研讨会会议录（美国加利福尼亚州圣迭戈）（POPL '98）。计算机协会，纽约，纽约州，美国，365–377。https://doi.org/10.1145/268946.268976 Shin-yaKatsumata。2018。分级线性指数余单子的双范畴论分析。载于《软件科学与计算结构基础》，Christel Baier 和 Ugo Dal Lago（编）。Springer International Publishing，Cham，110–127。Daniel Marshall、MichaelVollmer和Dominic Orchard。2022。线性性与唯一性：一项协约。载于《编程语言与系统》，Ilya Sergey（编）。Springer International Publishing，Cham，346–375。Conor McBride。2016。我有很多无物。Springer International Publishing，Cham，207–233。https://doi.org/10.1007/978-3-319-30936-1_12 Eugenio Moggi。1991。计算概念与单子。《信息与计算》93，1（1991），55–92。https://www.sciencedirect.com/science/article/pii/0890540191900524 选自 1989年 IEEE 计算机科学逻辑研讨会。Benjamin Moon、HarleyEadesIII和Dominic Orchard。2021。分级模态依赖类型论。载于《编程语言与系统》，Nobuko Yoshida（编）。Springer International Publishing，Cham，462–490。Andrew C.Myers。1999。JFlow：实用的主要静态信息流控制。载于第26 届ACMSIGPLAN-SIGACT 编程语言原理研讨会会议录（美国得克萨斯州圣安东尼奥）（POPL '99）。计算机协会，纽约，纽约州，美国，228–241。https://doi.org/10.1145/292540.292561 Dominic Orchard、Vilem-BenjaminLiepelt 和 HarleyEades III。2019年。使用分级模态类型进行定量程序推理。ACM 编程语言会议录 3，ICFP，文章 110（2019年7月），30 页。https://doi.org/10.1145/3341714 Tomas Petricek、DominicOrchard和Alan Mycroft。2014。共效应：上下文依赖计算的演算。载于国际函数式编程大会会议录（瑞典哥德堡）（ICFP2014）。AndrewM.Pitts。2013。名义集合：计算机科学中的名称与对称性。剑桥大学出版社，美国。FrédéricProst。2000。λ-立方体的依赖关系静态演算。载于第 15 届 IEEE计算机科学逻辑年会会议录（LICS '00）。IEEE 计算机学会，美国，267。NaokataShikuma和 Atsushi Igarashi。2006。通过完全完备翻译到简单类型 λ-演算证明非干扰。载于第 11 届亚洲计算机科学会议计算机科学进展：安全软件及相关问题会议录（日本东京）（ASIAN'06）。施普林格出版社，柏林，海德堡，301–315。Deian Stefan、David Mazières、John C.Mitchell 和 AlejandroRusso。2017。存在异常时的灵活动态信息流控制。《函数式编程杂志》27（2017），e5。https://doi.org/10.1017/S0956796816000241DavidN.Turner 和 Philip Wadler。1999。线性逻辑的操作解释。《理论计算机科学》227，1–2（1999年9月），231–248。https://doi.org/10.1016/S0304-3975(99)00054-7 Dennis Volpano、Cynthia Irvine 和 Geoffrey Smith。1996。用于安全流分析的可靠类型系统。《计算机安全杂志》4，2–3（1996年1月），167–187。Philip Wadler。1990。线性类型可以改变世界！载于《编程概念与方法》。North。

$$
\begin{aligned}
\text{grades, } q \in Q &::= 0 \mid 1 \mid q_1 + q_2 \mid q_1 \cdot q_2 \mid \ldots \\
\text{types, } A, B &::= \mathrm{Bool} \mid A \multimap B \mid !_q A \\
\text{terms, } a, b &::= x \mid \lambda x : A.b \mid b\ a \mid !_q\, a \mid \mathrm{let}\ !_q\, x\ \mathrm{be}\ a\ \mathrm{in}\ b \\
&\quad \mid \mathrm{true} \mid \mathrm{false} \mid \mathrm{if}\ b\ \mathrm{then}\ a_1\ \mathrm{else}\ a_2 \\
\text{contexts, } \Gamma &::= \emptyset \mid \Gamma, x :^q A
\end{aligned}
$$

图 11. GC(Q) 的文法

分级上下文类型系统中无法推导的合并

命题A.1。 分级上下文类型系统 *[Abel 和 Bernardy 2020年； Atkey 2018； Brunel 等人 2014； Choudhury 等人 2021； Ghica 和 Smith 2014； Orchard等人 2019年； Petricek 等人 2014]* 无法推导出单子连接算子。

证明。命题中提到的分级上下文类型系统在设计上略有差异，但它们都包含一个核心分级演算。下面，我们先介绍这一核心分级演算 GCore，随后证明 GCore 无法推导出单子连接算子。

GCore 由任意预序半环 $Q = (Q, +, \cdot, 0, 1 <:)$进行参数化。若以 Q作为参数，则该演算称为 GCore(Q)。 GCore(Q) 的文法和类型规则分别见图 11 和 12 。上下文上的运算定义与第 3.1 节相同。

$\boxed{\Gamma \vdash a : A}$ （类型规则）

$$\text{GC-变量}\ \frac{}{0 \cdot \Gamma_1, x :^1 A, 0 \cdot \Gamma_2 \vdash x : A}$$

$$\text{GC-抽象}\ \frac{\Gamma,\ x :^1 A \vdash b :\ B}{\Gamma \vdash \lambda x : A.\ b : A \multimap B}$$

$$\text{GC-应用}\ \frac{\Gamma_1 \vdash b : A \multimap B \qquad \Gamma_2 \vdash a : A \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash b\ a : B}$$

$$\text{GC-指数引入}\ \frac{\Gamma \vdash a : A}{q \cdot \Gamma \vdash !_q a :!_q A}$$

$$\text{GC-ExpElim}\ \frac{\Gamma_1 \vdash a :!_q A \qquad \Gamma_2,\ x :^q A \vdash b :\ B \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash \text{令 } !_q x \text{ 为 } a \text{ 中的 } b :\ B}$$

$$\text{GC-真值}\ \frac{}{0 \cdot \Gamma \vdash \text{真值} :\ \text{布尔值}}$$

$$\text{GC-False}\ \frac{}{0 \cdot \Gamma \vdash \mathrm{false} : \text{布尔值}}$$

$$\text{GC-If}\ \frac{\Gamma_1 \vdash b : \text{布尔值} \qquad \Gamma_2 \vdash a_1 : A \qquad \Gamma_2 \vdash a_2 : A \qquad \lfloor \Gamma_1 \rfloor = \lfloor \Gamma_2 \rfloor}{\Gamma_1 + \Gamma_2 \vdash \text{若 } b \text{ 则 } a_1 \text{ 否则 } a_2 : A}$$

$$\text{GC-Sub}\ \frac{\Gamma_1 \vdash a : A \qquad \Gamma_2 <: \Gamma_1}{\Gamma_2 \vdash a : A}$$

图12. GC(Q) 的类型规则

现在，为了导出矛盾，假设 GCore(Q) 能够推导出一个单子连接算子。那么，对于任意 A，存在一个类型为 $!_{q_1}\, !_{q_2} A \to !_{q_1 \cdot q_2} A$ 的闭的非常值函数，对所有 $q_1, q_2 \in Q$ 都成立。通过模型论论证，我们将证明，对于某个 Q、A、q_1 和 q_2，，任何具有此类型的函数都必须是常值函数。

令 Q 为以下预序半环。其底集为 $\{0, 1, k_1, k_2\}$，半环运算和预序定义如下：

$$q_1 + q_2 \triangleq k_2 \text{ if } q_1, q_2 \notin \{0\}$$
$$q_1 \cdot q_2 \triangleq k_2 \text{ if } q_1, q_2 \notin \{0, 1\}$$
$$q_1 <: q_2 \triangleq q_1 = q_2$$

检查上述定义是否满足预序半环的公理。现在，注意到 $\mathrm{k}_1 \cdot \mathrm{k}_1 = \mathrm{k}_2$。下文将证明，在 GCore(Q) 中不存在类型为 $!_{k_1}\,!_{k_1}\mathrm{Bool} \to !_{k_2}\mathrm{Bool}$ 的非恒定函数。

为得出矛盾，假设存在这样的函数，$\emptyset \vdash \mathrm{f} : !_{k_1}\,!_{k_1}$ 布尔值 $\to !_{k_2}$ 布尔值。于是，$\mathrm{f}\,(!_{k_1}\,!_{k_1}$ 真值) 和 $\mathrm{f}\,(!_{k_1}\,!_{k_1}\mathrm{false})$ 会归约为不同的值（假设采用按值调用的归约）。不失一般性，设 $\mathrm{f}\,(!_{k_1}\,!_{k_1}$真值) $\rightsquigarrow^* !_{k_2}$ 真值，且 $\mathrm{f}\,(!_{k_1}\,!_{k_1}\,\mathrm{false}) \rightsquigarrow^* !_{k_2}\mathrm{false}$，其中 $\rightsquigarrow^*$ 是多步按值调用归约关系。现在，对于 GCore(Q) 在模型 M 中的任意可靠解释，$[\![_]\!]_{\mathcal{M}}$，我们应有：

$$[\![f\ (!_{k_1}\,!_{k_1}\,\mathrm{true})]\!]_{\mathcal{M}} = [\![!_{k_2}\,\mathrm{true}]\!]_{\mathcal{M}} \text{ and } [\![f\ (!_{k_1}\,!_{k_1}\,\mathrm{false})]\!]_{\mathcal{M}} = [\![!_{k_2}\,\mathrm{false}]\!]_{\mathcal{M}}$$

We c构造一个 GCore(Q) 的可靠模型，使这些等式导致矛盾。

令 Set 为集合与函数构成的范畴。注意，Set 是一个对称幺半范畴，其幺半积为笛卡尔积。现在，任意一个作用于 $\mathcal{Q}$ 上的 Set 中的 GCore($\mathcal{Q}$) [Katsumata 2018] 提供了一个可靠解释。我们如下定义 !：一个作用于 $\mathcal{Q}$ 上的 Set 中的 Q 分级线性指数余单子：

$$!(0) = !(k_1) = *$$
$$!(1) = !(k_2) = \mathrm{Id}$$

这里，Id 是恒等函子，而 $*$ 是将每个对象映射到终对象的函子。与 ! 相关的态射符合预期。

现在，使用 ! 进行解释，可得：

$$[\![f\ (!_{k_1}\,!_{k_1}\,\mathrm{true})]\!]_{(\mathrm{Set},!)} = [\![f\ (!_{k_1}\,!_{k_1}\,\mathrm{false})]\!]_{(\mathrm{Set},!)} = \mathrm{app} \circ \langle [\![f]\!]_{(\mathrm{Set},!)}, \langle\rangle \rangle$$

但是 $[\![!_{k_2}$真值$]\!]_{(\mathrm{Set},!)} \neq [\![!_{k_2}$假$]\!]_{(\mathrm{Set},!)}$。这是一个矛盾。

□

B 简单类型 LDC 中的线性性分析

引理 B.1（乘法（引理 3.1））。若 $\Gamma \vdash \mathrm{a} :^{q} \mathrm{A}$，则 $\mathrm{r}_0 \cdot \Gamma \vdash \mathrm{a} :^{r_0 \cdot q} \mathrm{A}$。

证明。对 $\Gamma \vdash \mathrm{a} :^{q} \mathrm{A}$ 进行归纳。

- 规则 ST-Var。已知：$0 \cdot \Gamma_1,\ \mathrm{x} :^{q} \mathrm{A},\ 0 \cdot \Gamma_2 \vdash \mathrm{x} :^{q} \mathrm{A}$。需要证明：$0 \cdot \Gamma_1,\ \mathrm{x} :^{r_0 \cdot q} \mathrm{A},\ 0 \cdot \Gamma_2 \vdash \mathrm{x} :^{r_0 \cdot q} \mathrm{A}$。此情形由规则 ST-Var 得出。
- 规则 ST-Lam。已知：$\Gamma \vdash \lambda^{r}\mathrm{x} : \mathrm{A}.\mathrm{b} :^{q\ r} \mathrm{A} \to \mathrm{B}$，其中 $\Gamma,\ \mathrm{x} :^{q \cdot r} \mathrm{A} \vdash \mathrm{b} :^{q} \mathrm{B}$。需要证明：$\mathrm{r0} \cdot \Gamma \vdash \lambda^{r}\mathrm{x} : \mathrm{A}.\mathrm{b} :^{r_0 \cdot q\ r} \mathrm{A} \to \mathrm{B}$。根据归纳假设，$\mathrm{r0} \cdot \Gamma,\ \mathrm{x} :^{r_0 \cdot (q \cdot r)} \mathrm{A} \vdash \mathrm{b} :^{r_0 \cdot q} \mathrm{B}$。根据乘法的结合律，$\mathrm{r0} \cdot (\mathrm{q} \cdot \mathrm{r}) = (\mathrm{r0} \cdot \mathrm{q}) \cdot \mathrm{r}$。因此，此情形由规则 ST-Lam 得出。
- 规则 ST-App。已知：$\Gamma_1 + \Gamma_2 \vdash \mathrm{b}\ \mathrm{a}^{r} :^{q} \mathrm{B}$，其中 $\Gamma_1 \vdash \mathrm{b} :^{q\ r} \mathrm{A} \to \mathrm{B}$ 且 $\Gamma_2 \vdash \mathrm{a} :^{q \cdot r} \mathrm{A}$。需要证明：$\mathrm{r0} \cdot (\Gamma_1 + \Gamma_2) \vdash \mathrm{b}\ \mathrm{a}^{r} :^{r_0 \cdot q} \mathrm{B}$。根据归纳假设，$\mathrm{r0} \cdot \Gamma_1 \vdash \mathrm{b} :^{r_0 \cdot q\ r} \mathrm{A} \to \mathrm{B}$ 且 $\mathrm{r0} \cdot \Gamma_2 \vdash \mathrm{a} :^{r_0 \cdot (q \cdot r)} \mathrm{A}$。因此，此情形由规则 ST-App 得出，使用结合律和分配律性质。

• ST-Pair 规则。已知：$\Gamma_1 + \Gamma_2 \vdash (a_1{}^r, a2) :^q {}^r A_1 \times A2$，其中 $\Gamma_1 \vdash a1 :^{q\cdot r} A1$ 且 $\Gamma_2 \vdash a2 :^q A2$。需证明：$r0 \cdot (\Gamma_1 + \Gamma_2) \vdash (a_1{}^r, a2) :^{r_0 \cdot q} {}^r A_1 \times A2$。根据归纳假设，$r0 \cdot \Gamma_1 \vdash a1 :^{r_0 \cdot (q \cdot r)} A1$ 且 $r0 \cdot \Gamma_2 \vdash a2 :^{r_0 \cdot q} A2$。因此，此 case 表达式可由 ST-Pair 规则，并利用结合律和分配律性质得出。• 规则 ST-LetPair。已知：$\Gamma_1 + \Gamma_2 \vdash \text{let}^{q_0}(x\ {}^r, y)$ be a in b $:^q B$，其中 $\Gamma_1 \vdash a :^{q \cdot q_0} {}^r A_1 \times A2$ 且 $\Gamma_2, x :^{q \cdot q_0 \cdot r} A1, y :^{q \cdot q_0} A_2 \vdash b :^q B$。需证明：$r0 \cdot (\Gamma_1 + \Gamma_2) \vdash \text{let}^{q_0}(x\ {}^r, y)$ be a in b $:^{r_0 \cdot q} B$。根据归纳假设，$r0 \cdot \Gamma_1 \vdash a :^{r_0 \cdot (q \cdot q_0)} {}^r A_1 \times A2$ 且 $r0 \cdot \Gamma_2, x :^{r_0 \cdot (q \cdot q_0 \cdot r)} A1, y :^{r_0 \cdot (q \cdot q_0)} A_2 \vdash b :^{r_0 \cdot q} B$。因此，此 case 表达式可由 ST-LetPair 规则，并利用结合律和分配律性质得出。

• 规则 ST-Unit。已知：$0 \cdot \Gamma \vdash \text{unit} :^q$ 单元类型。需要证明：$r0 \cdot (0 \cdot \Gamma) \vdash \text{unit} :^{r_0 \cdot q}$ 单元类型。根据规则 ST-Unit，并利用结合律和单位元性质即可得到此情形。• 规则 ST-LetUnit。已知：$\Gamma_1 + \Gamma_2 \vdash \text{let}^{q_0}$ unit be a in b $:^q B$，其中 $\Gamma_1 \vdash a :^{q \cdot q_0}$ 单元类型，且 $\Gamma_2 \vdash b :^q B$。需要证明：$r0 \cdot (\Gamma_1 + \Gamma_2) \vdash \text{let}^{q_0}$unit be a in b $:^{r_0 \cdot q} B$。根据归纳假设，$r0 \cdot \Gamma_1 \vdash a :^{r_0 \cdot (q \cdot q_0)}$ 单元类型，且 $r0 \cdot \Gamma_2 \vdash b :^{r_0 \cdot q} B$。因此，根据规则 ST-LetUnit，并利用结合律和分配律性质即可得到此情形。

• 规则 ST-Inj1。设 $\Gamma \vdash \text{inj1 a1} :^q A_1 + A2$，其中 $\Gamma \vdash a1 :^q A1$。需要证明：$r0 \cdot \Gamma \vdash \text{inj1 a1} :^{r_0 \cdot q} A_1 + A2$。根据归纳假设，$r0 \cdot \Gamma \vdash a1 :^{r_0 \cdot q} A1$。因此，此 case 表达式由规则 ST-Inj1 得出。

• 规则ST-Inj2。类似于规则 ST-Inj1。• 规则 ST-Case。已知：$\Gamma_1 + \Gamma_2 \vdash \text{case}$表达式^{q_0} aofx1.b1;x2.b2$:^q B$，其中 $\Gamma_1 \vdash a:^{q \cdot q_0} A_1 + A2$且 Γ_2, $x1 :^{q \cdot q_0} A_1 \vdash b1 :^q B$，以及 Γ_2,$x2 :^{q \cdot q_0} A_2 \vdash b2 :^q B$。需要证明：$r0 \cdot (\Gamma_1 + \Gamma_2) \vdash \text{case}$ 表达式^{q_0}a ofx1.b1 ; x2.b2:$^{r_0 \cdot q} B$。根据归纳假设，$r0 \cdot \Gamma_1 \vdash a :^{r_0 \cdot (q \cdot q_0)} A_1 + A2$且 $r0 \cdot \Gamma_2$,$x1:^{r_0 \cdot (q \cdot q_0)} A_1 \vdash b1 :^{r_0 \cdot q} B$，以及 $r0 \cdot \Gamma_2$,$x2:^{r_0 \cdot (q \cdot q_0)} A_2 \vdash b2 :^{r_0 \cdot q} B$。因此，该情形可由规则 ST-Case，利用结合律和分配律性质得出。

• 规则 ST-SubL。已知：$\Gamma \vdash a :^q A$，其中 $\Gamma' \vdash a :^q A$ 且 $\Gamma <: \Gamma'$。需要证明：$r0 \cdot \Gamma \vdash a :^{r_0 \cdot q} A$。根据归纳假设，$r0 \cdot \Gamma' \vdash a :^{r_0 \cdot q} A$。接下来，由于 $\Gamma <: \Gamma'$，有 $r0 \cdot \Gamma <: r0 \cdot \Gamma'$。因此，本 case 表达式可由规则 ST-SubL 得出。
• 规则 ST-SubR。已知：$\Gamma \vdash a :^{q'} A$，其中 $\Gamma \vdash a :^q A$ 且 $q <: q'$。需要证明：$r0 \cdot \Gamma \vdash a :^{r_0 \cdot q'} A$。根据归纳假设，$r0 \cdot \Gamma \vdash a :^{r_0 \cdot q} A$。接下来，由于 $q <: q'$，有 $r0 \cdot q <: r0 \cdot q'$。因此，本 case 表达式可由规则 ST-SubR 得出。

□

引理B.2（因子分解（引理 3.2））。如果 $\Gamma \vdash a :^q A$ 且 $q \neq 0$，则存在 Γ' 使得$\Gamma' \vdash a :^1 A$且 $\Gamma <: q \cdot \Gamma'$。

证明。根据 $\Gamma \vdash a :^q A$ 进行归纳。

• 规则 ST-Var。已知：$0 \cdot \Gamma_1, x :^q A, 0 \cdot \Gamma_2 \vdash x :^q A$。需要证明：存在 $\exists \Gamma'$，使得 $\Gamma' \vdash x :^1 A$，且 $0 \cdot \Gamma_1, x :^q A, 0 \cdot \Gamma_2 <: q \cdot \Gamma'$。此 case 表达式通过令 $\Gamma' := 0 \cdot \Gamma_1, x :^1 A, 0 \cdot \Gamma_2$ 即可得到。

• 规则 ST-Lam。已知：$\Gamma \vdash \lambda^{r} x : A.b :^{q\ r} A \to B$，其中 Γ，$x :^{q \cdot r} A \vdash b :^{q} B$。需要证明：存在 $\exists \Gamma'$，使得 $\Gamma' \vdash \lambda^{r} x : A.b :^{1\ r} A \to B$，且 $\Gamma <: q \cdot \Gamma'$。根据归纳假设，存在 $\exists \Gamma' 1$ 和 $r' 1$，使得 $\Gamma' 1$，$x :^{r' 1} A \vdash b :^{1} B$，且 $\Gamma <: q \cdot \Gamma' 1$ 和 $q \cdot r <: q \cdot r' 1$。由于 $q \cdot r <: q \cdot r' 1$，且 $q \neq 0$，因此 $r <: r' 1$。根据规则 ST-SubL，$\Gamma' 1$，$x :^{r} A \vdash b :^{1} B$。于是，此 case 表达式根据规则 ST-Lam，令 $\Gamma' := \Gamma' 1$ 即可得到。• 规则 ST-App。已知：$\Gamma_1 + \Gamma_2 \vdash b\ a^{r} :^{q} B$，其中 $\Gamma_1 \vdash b :^{q\ r} A \to B$ 且 $\Gamma_2 \vdash a :^{q \cdot r} A$。需要证明：存在 $\exists \Gamma'$，使得 $\Gamma' \vdash b\ a^{r} :^{1} B$，且 $\Gamma_1 + \Gamma_2 <: q \cdot \Gamma'$。需要考虑两种情况。– $r \neq 0$。由于 $q \neq 0$，因此 $q \cdot r \neq 0$。根据归纳假设，存在 $\exists \Gamma' 1$、$\Gamma' 2$，使得 $\Gamma'_1 \vdash b :^{1\ r} A \to B$，$\Gamma'_2 \vdash a :^{1} A$，且 $\Gamma_1 <: q \cdot \Gamma' 1$ 和 $\Gamma_2 <: (q \cdot r) \cdot \Gamma' 2$。根据引理 B.1，$r \cdot \Gamma'_2 \vdash a :^{r} A$。根据规则 ST-App，$\Gamma'_1 + r \cdot \Gamma'_2 \vdash b\ a^{r} :^{1} B$。于是，此 case 表达式通过令 $\Gamma' := \Gamma'_1 + r \cdot \Gamma' 2$ 即可得到。– $r = 0$。于是，$\Gamma_2 \vdash a :^{0} A$。根据引理 B.1，$0 \cdot \Gamma_2 \vdash a :^{0} A$。根据归纳假设，存在 $\exists \Gamma' 1$，使得 $\Gamma'_1 \vdash b :^{1\ 0} A \to B$，且 $\Gamma_1 <: q \cdot \Gamma' 1$。根据规则 ST-App，$\Gamma'_1 \vdash b\ a^{0} :^{1} B$。此外，$\Gamma_1 + \Gamma_2 <: \Gamma_1$（$\because \Gamma_1 + \Gamma_0 <: \Gamma_1$ 对于 Γ_0 in $\mathbb{N}_{\geq}$ 和 $\Gamma_2 = 0$ in $\mathbb{N}_{=}$ 中的任意者）。于是，此 case 表达式通过令 $\Gamma' := \Gamma' 1$ 即可得到。• 规则 ST-Pair。已知：$\Gamma_1 + \Gamma_2 \vdash (a_1^{r}, a2) :^{q\ r} A_1 \times A2$，其中 $\Gamma_1 \vdash a1 :^{q \cdot r} A1$ 且 $\Gamma_2 \vdash a2 :^{q} A2$。需要证明：存在 $\exists \Gamma'$，使得 $\Gamma' \vdash (a_1^{r}, a2) :^{1\ r} A_1 \times A2$，且 $\Gamma_1 + \Gamma_2 <: q \cdot \Gamma'$。需要考虑两种情况。– $r \neq 0$。由于 $q \neq 0$，因此 $q \cdot r \neq 0$。根据归纳假设，存在 $\exists \Gamma' 1$、$\Gamma' 2$，使得 $\Gamma'_1 \vdash a1 :^{1} A1$，$\Gamma'_2 \vdash a2 :^{1} A2$，且 $\Gamma_1 <: (q \cdot r) \cdot \Gamma' 1$ 和 $\Gamma_2 <: q \cdot \Gamma' 2$。根据引理 B.1，$r \cdot \Gamma'_1 \vdash a1 :^{r} A1$。根据 ST-Pair 规则，$r \cdot \Gamma'_1 + \Gamma'_2 \vdash (a_1^{r}, a2) :^{1\ r} A_1 \times A2$。于是，此 case 表达式通过令 $\Gamma' := r \cdot \Gamma'_1 + \Gamma' 2$ 即可得到。– $r = 0$。于是，$\Gamma_1 \vdash a1 :^{0} A1$。根据引理 B.1，$0 \cdot \Gamma_1 \vdash a1 :^{0} A1$。根据归纳假设，存在 $\exists \Gamma' 2$，使得 $\Gamma'_2 \vdash a2 :^{1} A2$，且 $\Gamma_2 <: q \cdot \Gamma' 2$。根据 ST-Pair 规则，$\Gamma'_2 \vdash (a10, a2) :^{1\ 0} A_1 \times A2$。此外，$\Gamma_1 + \Gamma_2 <: \Gamma_2$（$\because \Gamma_0 + \Gamma_2 <: \Gamma_2$ 对于 Γ_0 in $\mathbb{N}_{\geq}$ 和 $\Gamma_1 = \overline{0}$ in $\mathbb{N}_{=}$ 中的任意者）。于是，此 case 表达式通过令 $\Gamma' := \Gamma' 2$ 即可得到。• 规则 ST-LetPair。已知：$\Gamma_1 + \Gamma_2 \vdash \mathrm{let}^{q_0} (x^{r}, y)$ be a in b $:^{q} B$，其中 $\Gamma_1 \vdash a :^{q \cdot q_0\ r} A_1 \times A2$，且 Γ_2，$x :^{q \cdot {}_{q_0} \cdot r} A1$，$y :^{q \cdot q_0} A_2 \vdash b :^{q} B$，且 $q_0 <: 1$。需要证明：存在 $\exists \Gamma'$，使得 $\Gamma' \vdash \mathrm{let}^{q_0} (x^{r}, y)$ be a in b $:^{1} B$，且 $\Gamma_1 + \Gamma_2 <: q \cdot \Gamma'$。由于 $q \neq 0$ 且 $q_0 <: 1$，因此 $q \cdot q_0 \neq 0$。根据归纳假设，存在 $\exists \Gamma' 1$、$\Gamma' 2$ 和 r'、q'，使得 $\Gamma'_1 \vdash a :^{1\ r} A_1 \times A2$，$\Gamma' 2$，$x :^{r'} A1$，$y :^{q'} A_2 \vdash b :^{1} B$，且 $\Gamma_1 <: (q \cdot q0) \cdot \Gamma' 1$，$\Gamma_2 <: q \cdot \Gamma' 2$，$q \cdot q0 \cdot r <: q \cdot r'$，以及 $q \cdot q_0 <: q \cdot q'$。由于 $q \neq 0$，因此 $q0 \cdot r <: r'$ 且 $q_0 <: q'$。根据规则 ST-SubL，$\Gamma' 2$，$x :^{q_0 \cdot r} A1$，$y :^{q_0} A_2 \vdash b :^{1} B$。再次根据引理 B.1，$q0 \cdot \Gamma'_1 \vdash a :^{q_0\ r} A_1 \times A2$。根据规则 ST-LetPair，$q0 \cdot \Gamma'_1 + \Gamma'_2 \vdash \mathrm{let}^{q_0} (x^{r}, y)$ be a in b $:^{1} B$。于是，此 case 表达式通过令 $\Gamma' := q0 \cdot \Gamma'_1 + \Gamma' 2$ 即可得到。

• 规则ST-Unit。给定：$0 \cdot \Gamma \vdash$ 单元类型 $:^{q}$ 单元类型。需证明：$\exists \Gamma'$，使得 $\Gamma' \vdash$ 单元类型 $:^{1}$ 单元类型且 $0 \cdot \Gamma <: q \cdot \Gamma'$。此情形可通过令 $\Gamma' := 0 \cdot \Gamma$ 得到。

• 规则 ST-LetUnit。已知：$\Gamma_1 + \Gamma_2 \vdash \text{let}_{q_0}\ \text{unitbeain b}:^{q} B$，其中 $\Gamma_1 \vdash a :^{q\cdot q_0}$ 单元类型，且 $\Gamma_2 \vdash b :^{q} B$，以及 $q_0 <: 1$。需要证明：存在 $\exists\Gamma'$，使得 $\Gamma' \vdash \text{let}_{q_0}\ \text{unit be ain b}:^{1} B$且 $\Gamma_1 + \Gamma_2 <: q\cdot\Gamma'$。根据归纳假设，存在 $\exists\Gamma'1, \Gamma'2$，使得 $\Gamma'_1 \vdash a:^{1}$ 单元类型，且 $\Gamma'_2 \vdash b :^{1} B$，以及 $\Gamma_1 <: (q\cdot q0)\cdot\Gamma'1$和 $\Gamma_2 <: q\cdot\Gamma'2$。根据引理 B.1，$q0\cdot\Gamma'_1 \vdash a :^{q_0}$ 单元类型。根据规则 ST-Let Unit，$q0\cdot\Gamma'_1 + \Gamma'_2 \vdash \text{let}_{q_0}\ \text{unitbea in b}:^{1} B$。该 case表达式可通过令 $\Gamma' := q0\cdot\Gamma'_1 + \Gamma'2$ 得出。

• 规则ST-Inj1。设 $\Gamma \vdash \text{inj1a1}:^{q} A_1 + A2$，其中 $\Gamma \vdash a1:^{q} A1$。需证明：存在 $\exists\Gamma'$，使得 $\Gamma' \vdash \text{inj1 a1} :^{1} A_1 + A2$ 且 $\Gamma <: q\cdot\Gamma'$。根据归纳假设，存在 $\exists\Gamma'1$，使得 $\Gamma'_1 \vdash a1 :^{1} A1$ 且 $\Gamma <: q\cdot\Gamma'1$。通过令 $\Gamma' := \Gamma'1$ 即可得到此 case 表达式。

• 规则 ST-Inj2。与规则 ST-Inj1 类似。 • 规则 ST-Case。已知：$\Gamma_1 + \Gamma_2 \vdash \text{case}_{q_0}\ a\ \text{of x1.b1 ; x2.b2} :^{q} B$，其中 $\Gamma_1 \vdash a :^{q\cdot q_0} A_1 + A2$，且 $\Gamma_2, x1 :^{q\cdot q_0} A_1 \vdash b1 :^{q} B$，且 $\Gamma_2, x2 :^{q\cdot q_0} A_2 \vdash b2 :^{q} B$，且 $q_0 <: 1$。需证明：存在 $\exists\Gamma'$，使得 $\Gamma' \vdash \text{case}_{q_0}\ a\ \text{of x1.b1 ; x2.b2} :^{1} B$，且 $\Gamma_1 + \Gamma_2 <: q\cdot\Gamma'$。根据归纳假设，存在 $\exists\Gamma'1, \Gamma'21, \Gamma'22$ 和 $q'1, q'2$，使得 $\Gamma'_1 \vdash a :^{1} A_1 + A2$，且 $\Gamma'21, x1 :^{q'_1} A_1 \vdash b1 :^{1} B$，且 $\Gamma'22, x2 :^{q'_2} A_2 \vdash b2 :^{1} B$，且 $\Gamma_1 <: (q\cdot q0)\cdot\Gamma'1$，且 $\Gamma_2 <: q\cdot\Gamma'21$，且 $\Gamma_2 <: q\cdot\Gamma'22$，且 $q\cdot q_0 <: q\cdot q'1$，且 $q\cdot q_0 <: q\cdot q'2$。由于 $q \neq 0$，因此 $q_0 <: q'1$，且 $q_0 <: q'2$。此 case 表达式的证明其余部分对于 $\mathbb{N}_=$ 和 $\mathbb{N}_\geq$ 有所不同。– 在 $\mathbb{N}_=$ 的情况下，由于 $<:$ 是离散的，$\Gamma'_{21} = \Gamma'_{22} = \Gamma'2$（设为此）。于是，$\Gamma_2 <: q\cdot\Gamma'2$。此外，根据规则 ST-SubL，$\Gamma'2, x1 :^{q_0} A_1 \vdash b1 :^{1} B$，且 $\Gamma'2, x2 :^{q_0} A_2 \vdash b2 :^{1} B$。根据引理 B.1，$q0\cdot\Gamma'_1 \vdash a :^{q_0} A_1 + A2$。于是，此情形可通过令 $\Gamma' := q0\cdot\Gamma'_1 + \Gamma'2$ 得证。– 在 $\mathbb{N}_\geq$的情况下，$\Gamma'21$ 可能不等于 $\Gamma'22$。因此我们需要采用以下构造。对于数量 q1、q2，定义一个运算符：$q1; q_2 =$若 $q_1 <: q2$，则为 q1；否则为 q2。将该运算符扩展到上下文 Γ_1 和 Γ_2（其中 $\lfloor\Gamma_1\rfloor = \lfloor\Gamma_2\rfloor$），定义 $\Gamma_0 := \Gamma_1; \Gamma_2$ 为 $\lfloor\Gamma_0\rfloor = \lfloor\Gamma_1\rfloor$ 和 $\Gamma_0 = \underline{\Gamma_1}; \underline{\Gamma_2}$（逐点定义）。现在，令 $\Gamma'2 := \Gamma'21; \Gamma'22$。于是，$\Gamma_2 <: q\cdot\Gamma'2$。同样，根据规则 ST-SubL，$\Gamma'2, x1 :^{q_0} A_1 \vdash b1 :^{1} B$，且 $\Gamma'2, x2 :^{q_0} A_2 \vdash b2 :^{1} B$。根据引理 B.1，$q0\cdot\Gamma'_1 \vdash a :^{q_0} A_1 + A2$。于是，此情形可通过令 $\Gamma' := q0\cdot\Gamma'_1 + \Gamma'2$ 得证。

• 规则 ST-SubL。已知：$\Gamma \vdash a :^{q} A$，其中 $\Gamma_1 \vdash a :^{q} A$ 且 $\Gamma <: \Gamma_1$。需要证明：存在 $\exists\Gamma_\prime$，使得 $\Gamma_\prime \vdash a :^{1} A$ 且 $\Gamma <: q\cdot\Gamma_\prime$。由归纳假设，存在 $\exists\Gamma_{\prime 1}$，使得 $\Gamma_{\prime 1} \vdash a :^{1} A$ 且 $\Gamma_1 <: q\cdot\Gamma_{\prime 1}$。因此，令 $\Gamma_\prime := \Gamma_{\prime 1}$，即可得证。

• 规则 ST-SubR。已知：$\Gamma \vdash a :^{q'} A$，其中 $\Gamma \vdash a :^{q} A$ 且 $q <: q'$。需要证明：存在 $\exists\Gamma'$，使得 $\Gamma' \vdash a :^{1} A$ 且 $\Gamma <: q'\cdot\Gamma'$。由于 $q' \neq 0$，因此 $q \neq 0$。由归纳假设，存在 $\exists\Gamma'1$，使得 $\Gamma'_1 \vdash a :^{1} A$ 且 $\Gamma <: q\cdot\Gamma'1$。现在，由于 $q <: q'$，因此 $q\cdot\Gamma'_1 <: q'\cdot\Gamma'1$。于是，令 $\Gamma' := \Gamma'1$，即可得证。

□

引理 B.3（拆分（引理 3.3））。 如果 $\Gamma \vdash a :\ ^{q_1+q_2} A$，则存在 Γ_1 和 Γ_2 使得$\Gamma_1 \vdash a :\ ^{q_1} A$ 以及 $\Gamma_2 \vdash a :\ ^{q_2} A$ 以及 $\Gamma = \Gamma_1 + \Gamma_2$。

证明。若 $q_1 + q_2 = 0$，则 $\Gamma_1 := 0 \cdot \Gamma$ and $\Gamma_2 := \Gamma$。否则，根据引理 B.2，$\exists \Gamma'$ 使得 $\Gamma' \vdash a :^1 A$ and $\Gamma <: (q_1 + q_2) \cdot \Gamma'$。于是，对于某个 Γ_0，有 $\Gamma = (q_1 + q_2) \cdot \Gamma' + \Gamma_0$（若是 $\mathbb{N}_=$ 的情况，则有 $\Gamma_0 = \overline{0}$）。现在，根据引理 B.1，$q_1 \cdot \Gamma' \vdash a :^{q_1} A$ and $q_2 \cdot \Gamma' \vdash a :^{q_2} A$。根据规则 ST-SubL，$q_2 \cdot \Gamma' + \Gamma_0 \vdash a :^{q_2} A$。令 $\Gamma_1 := q_1 \cdot \Gamma'$ 且 $\Gamma_2 := q_2 \cdot \Gamma' + \Gamma_0$，即可得出该引理。 □

引理 B.4（弱化（引理 3.4））。若 $\Gamma_1, \Gamma_2 \vdash a : {}^{q}A$，则 $\Gamma_1, z : {}^{0}C\ \Gamma_2 \vdash a : {}^{q}A$。

证明。对 $\Gamma_1, \Gamma_2 \vdash a :^q A$ 进行归纳。 □

引理 B.5（替换（引理 3.5））。若 $\Gamma_1, z :^{r_0} C\ \Gamma_2 \vdash a :^q A$ 且 $\Gamma \vdash c :^{r_0} C$ 且 $\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$，则 $\Gamma_1 + \Gamma,\ \Gamma_2 \vdash a\{c/z\} :^q A$。

证明。对 $\Gamma_1, z :^{r_0} C\ \Gamma_2 \vdash a :^q A$. 做归纳。

- 规则ST-Var。有三种情况需要考虑。–$0 \cdot \Gamma_{11}$，$z :^0 C$，$0 \cdot \Gamma_{12}$，$x :^q A$，$0 \cdot \Gamma_2 \vdash x :^q A$。此外，$\Gamma \vdash c :^0 C$，其中 $\lfloor \Gamma_{11} \rfloor = \lfloor \Gamma \rfloor$。需要证明：$\Gamma$，$0 \cdot \Gamma_{12}$，$x :^q A$，$0 \cdot \Gamma_2 \vdash x :^q A$。在 $\mathbb{N}_=$ 的 case 表达式中，我们有 $\Gamma = \overline{0}$。因此，这种情况由规则ST-Var 得出。在 $\mathbb{N}_\geq$ 的case 表达式中，这种情况由规则 ST-Var和规则ST-SubL 得出。– $0 \cdot \Gamma_1$，$x :^q A$，$0 \cdot \Gamma_2 \vdash x :^q A$。此外，$\Gamma \vdash a :^q A$，其中 $\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$。需要证明：$\Gamma$，$0 \cdot \Gamma_2 \vdash a :^q A$。由引理 B.4 得出。– $0 \cdot \Gamma_1$，$x :^q A$，$0 \cdot \Gamma_{21}$，$z :^0 C$，$0 \cdot \Gamma_{22} \vdash x :^q A$。此外，$\Gamma_{31}$，$x :^r A\ \Gamma_{32} \vdash c :^0 C$，其中 $\lfloor \Gamma_{31} \rfloor = \lfloor \Gamma_1 \rfloor$ 且 $\lfloor \Gamma_{32} \rfloor = \lfloor \Gamma_{21} \rfloor$。需要证明：$\Gamma_{31}$，$x :^{(q+r)} A\ \Gamma_{32}$，$0 \cdot \Gamma_{22} \vdash x :^q A$。在 $\mathbb{N}_=$ 的 case 表达式中，我们有 $\Gamma_{31} = 0$ 且 $\Gamma_{32} = 0$ 且 $r = 0$。因此，这种情况由规则 ST-Var 得出。在 $\mathbb{N}_\geq$ 的case 表达式中，这种情况由规则 ST-Var和规则 ST-SubL得出。
- 规则ST-Lam。已知：Γ_1，$z :^{r_0} C\ \Gamma_2 \vdash \lambda^r x : A.b :^{q\ r} A \to B$，其中 Γ_1，$z :^{r_0} C\ \Gamma_2$，$x :^{q \cdot r} A \vdash b :^q B$。此外，$\Gamma \vdash c :^{r_0} C$，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_1 \rfloor$。需要证明：$\Gamma_1 + \Gamma, \Gamma_2 \vdash \lambda^r x : A.b\{c/z\} :^{q\ r} A \to B$。由归纳假设和规则 ST-Lam 得出。

- 规则 ST-App。已知：$\Gamma_{11} + \Gamma_{12}, z :^{r_{01}+r_{02}} C\ \ \Gamma_{21} + \Gamma_{22} \vdash b\ a^r :^q B$，其中 $\Gamma_{11}, z :^{r_{01}} C\ \ \Gamma_{21} \vdash b :^{q\ r} A \to B$，且 $\Gamma_{12}, z :^{r_{02}} C\ \ \Gamma_{22} \vdash a :^{q \cdot r} A$。另有 $\Gamma \vdash c :^{r_{01}+r_{02}} C$，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_{11} \rfloor$。需要证明：$\Gamma_{11} + \Gamma_{12} + \Gamma, \Gamma_{21} + \Gamma_{22} \vdash b\{c/z\}\ a\{c/z\}^r :^q B$。根据引理 B.3，$\exists \Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c :^{r_{01}} C$ 且 $\Gamma_{32} \vdash c :^{r_{02}} C$，并且 $\Gamma = \Gamma_{31} + \Gamma_{32}$。根据归纳假设，$\Gamma_{11} + \Gamma_{31}, \Gamma_{21} \vdash b\{c/z\} :^{q\ r} A \to B$，且 $\Gamma_{12} + \Gamma_{32}, \Gamma_{22} \vdash a\{c/z\} :^{q \cdot r} A$。因此，这种情况可由规则 ST-App 得出。
- ST-Pair 规则。已知：$\Gamma_{11} + \Gamma_{12}, z :^{r_{01}+r_{02}} C\ \ \Gamma_{21} + \Gamma_{22} \vdash (a_1^r, a2) :^{q\ r} A_1 \times A2$，其中 $\Gamma_{11}, z :^{r_{01}} C\ \ \Gamma_{21} \vdash a1 :^{q \cdot r} A1$，且 $\Gamma_{12}, z :^{r_{02}} C\ \ \Gamma_{22} \vdash a2 :^q A2$。此外，$\Gamma \vdash c :^{r_{01}+r_{02}} C$，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_{11} \rfloor$。需证明：$\Gamma_{11} + \Gamma_{12} + \Gamma, \Gamma_{21} + \Gamma_{22} \vdash (a1\{c/z\}^r, a2\{c/z\}) :^{q\ r} A_1 \times A2$。由引理 B.3，$\exists \Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c :^{r_{01}} C$ 且 $\Gamma_{32} \vdash c :^{r_{02}} C$，并且 $\Gamma = \Gamma_{31} + \Gamma_{32}$。根据归纳假设，$\Gamma_{11} + \Gamma_{31}, \Gamma_{21} \vdash a1\{c/z\} :^{q \cdot r} A1$，且 $\Gamma_{12} + \Gamma_{32}, \Gamma_{22} \vdash a2\{c/z\} :^q A2$。因此，这种情况可由 ST-Pair 规则推出。

- 规则 ST-LetPair。已知：$\Gamma_{11} + \Gamma_{12},\ z :^{r_{01}+r_{02}} C\ \ \Gamma_{21} + \Gamma_{22} \vdash \text{let}_{q_0}(x^{\ r},\ y)\ \text{be a in b} :^q B$，其中 $\Gamma_{11},\ z :^{r_{01}} C\ \ \Gamma_{21} \vdash a :^{q \cdot q_0\ r} A_1 \times A_2$ 且 $\Gamma_{12},\ z :^{r_{02}} C\ \ \Gamma_{22},\ x :^{q \cdot q_0 r} A_1,\ y :^{q \cdot q_0} A_2 \vdash b :^q B$。此外，$\Gamma \vdash c :^{r_{01}+r_{02}} C$，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_{11} \rfloor$。

需要证明：$\Gamma_{11}+\Gamma_{12}+\Gamma, \Gamma_{21}+\Gamma_{22} \vdash \text{let}_{q_0}(x\ ^{r}, y)$ be a{c/z } in b{c/z } $:^{q} B$ 。根据引理 B.3，$\exists \Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c:^{r_{01}} C$ 且 $\Gamma_{32} \vdash c:^{r_{02}} C$，并且 $\Gamma=\Gamma_{31}+\Gamma_{32}$。根据归纳假设，$\Gamma_{11}+\Gamma_{31}, \Gamma_{21} \vdash a\{c/z\}:^{q \cdot q_0}\ ^{r} A_1 \times A_2$ 以及 $\Gamma_{12}+\Gamma_{32}, \Gamma_{22}, x:^{q \cdot q_0 \cdot r} A_1, y:^{q \cdot q_0} A_2 \vdash b\{c/z\}:^{q} B$ 。因此，这个 case 表达式可由规则 ST-LetPair 推出。

- 规则 ST-Unit。已知：$0 \cdot \Gamma_1$，$z:^{0} C$，$0 \cdot \Gamma_2 \vdash \text{unit}:^{q}$ 单元类型。另外 $\Gamma \vdash c:^{0} C$，其中 $\lfloor\Gamma\rfloor=\lfloor\Gamma_1\rfloor$。需证明：$\Gamma$，$0 \cdot \Gamma_2 \vdash \text{unit}:^{q}$ 单元类型。在 $\mathbb{N}_{=}$ 的 case 表达式中，我们有，$\Gamma=\overline{0}$。因此，此 case 表达式可由规则 ST-Unit 得出。在 $\mathbb{N}_{\geq}$ 的 case 表达式中，此 case 表达式可由规则 ST-Unit 和规则 ST-SubL 得出。

- 规则ST-LetUnit。已知：$\Gamma_{11}+\Gamma_{12}, z:^{r_{01}+r_{02}} C$ $\Gamma_{21}+\Gamma_{22} \vdash \text{let}_{q_0}$ unitbe ainb $:^{q} B$，其中 Γ_{11}，$z:^{r_{01}} C$ $\Gamma_{21} \vdash a:^{q \cdot q_0}$ 单元类型且 $\Gamma_{12}, z:^{r_{02}} C$ $\Gamma_{22} \vdash b:^{q} B$。此外，$\Gamma \vdash c:^{r_{01}+r_{02}} C$，其中 $\lfloor\Gamma\rfloor=\lfloor\Gamma_{11}\rfloor$。需证明：$\Gamma_{11}+\Gamma_{12}+\Gamma, \Gamma_{21}+\Gamma_{22} \vdash \text{let}_{q_0}$ unit be a{c/z} in b{c/z} $:^{q} B$。根据引理 B.3，$\exists \Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c:^{r_{01}} C$ 且 $\Gamma_{32} \vdash c:^{r_{02}} C$，并且 $\Gamma=\Gamma_{31}+\Gamma_{32}$。根据归纳假设，$\Gamma_{11}+\Gamma_{31}$，$\Gamma_{21} \vdash a\{c/z\}:^{q \cdot q_0}$ 单元类型 且 $\Gamma_{12}+\Gamma_{32}, \Gamma_{22} \vdash b\{c/z\}:^{q} B$。因此，此 case 表达式由规则 ST-LetUnit 得出。
- 规则 ST-Inj1。已知：Γ_1、$z:^{r_0} C$ $\Gamma_2 \vdash$ inj1a1$:^{q} A_1+A2$，其中 Γ_1、$z:^{r_0} C$ $\Gamma_2 \vdash a1:^{q} A1$。此外，$\Gamma \vdash c:^{r_0} C$，其中 $\lfloor\Gamma\rfloor=\lfloor\Gamma_1\rfloor$。需要证明：$\Gamma_1+\Gamma, \Gamma_2 \vdash$ inj1a1{c/z} $:^{q} A_1+A2$。根据归纳假设，$\Gamma_1+\Gamma, \Gamma_2 \vdash a1\{c/z\}:^{q} A1$。于是，这个 case 表达式可由规则 ST-Inj1 得出。

- 规则 ST-Inj2。类似于规则 ST-Inj1。
- 规则 ST-Case。已知：$\Gamma_{11}+\Gamma_{12}$，$z:^{r_{01}+r_{02}} C$ $\Gamma_{21}+\Gamma_{22} \vdash \text{case}_{q_0}$ a of x1.b1 ; x2.b2 $:^{q} B$，其中 Γ_{11}，$z:^{r_{01}} C$ $\Gamma_{21} \vdash a:^{q \cdot q_0} A_1+A2$，且 Γ_{12}，$z:^{r_{02}} C$ Γ_{22}，$x1:^{q \cdot q_0} A_1 \vdash b1:^{q} B$，且 Γ_{12}，$z:^{r_{02}} C$ Γ_{22}，$x2:^{q \cdot q_0} A_2 \vdash b2:^{q} B$。此外，$\Gamma \vdash c:^{r_{01}+r_{02}} C$，其中 $\lfloor\Gamma\rfloor=\lfloor\Gamma_{11}\rfloor$。需要证明：$\Gamma_{11}+\Gamma_{12}+\Gamma, \Gamma_{21}+\Gamma_{22} \vdash \text{case}_{q_0}$ a{c/z } of x1.b1{c/z } ; x2.b2{c/z } $:^{q} B$。根据引理 B.3，$\exists \Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c:^{r_{01}} C$ 且 $\Gamma_{32} \vdash c:^{r_{02}} C$，并且 $\Gamma=\Gamma_{31}+\Gamma_{32}$。根据归纳假设，$\Gamma_{11}+\Gamma_{31}, \Gamma_{21} \vdash a\{c/z\}:^{q \cdot q_0} A_1+A2$，且 $\Gamma_{12}+\Gamma_{32}, \Gamma_{22}$，$x1:^{q \cdot q_0} A_1 \vdash b1\{c/z\}:^{q} B$，且 $\Gamma_{12}+\Gamma_{32}, \Gamma_{22}, x2:^{q \cdot q_0} A_2 \vdash b2\{c/z\}:^{q} B$。因此，这种情况可由规则 ST-Case 得出。
- 规则 ST-SubL。已知：Γ_1，$z:^{r_0} C$ $\Gamma_2 \vdash a:^{q} A$，其中 $\Gamma' 1, z:^{r'_0} C$ $\Gamma'_2 \vdash a:^{q} A$，其中 $\Gamma_1 <: \Gamma' 1$ 且 $r_0 <: r' 0$ 且 $\Gamma_2 <: \Gamma' 2$。此外，$\Gamma \vdash c:^{r_0} C$，其中 $\lfloor\Gamma\rfloor=\lfloor\Gamma_1\rfloor$。需要证明：$\Gamma_1+\Gamma$，$\Gamma_2 \vdash a\{c/z\}:^{q} A$。由于 $r_0 <: r' 0$，根据规则 ST-SubR，$\Gamma \vdash c:^{r'_0} C$ 。根据归纳假设，$\Gamma'_1+\Gamma, \Gamma'_2 \vdash a\{c/z\}:^{q} A$。因此，本 case 表达式可由规则 ST-SubL 得出。

- 规则 ST-SubR。已知：$\Gamma_1, z:^{r_0} C$ $\Gamma_2 \vdash a:^{q'} A$，其中 $\Gamma_1, z:^{r_0} C$ $\Gamma_2 \vdash a:^{q} A$ 且 $q <: q'$。此外，$\Gamma \vdash c:^{r_0} C$，其中 $\lfloor\Gamma\rfloor=\lfloor\Gamma_1\rfloor$。需要证明：$\Gamma_1+\Gamma, \Gamma_2 \vdash a\{c/z\}:^{q'} A$。根据归纳假设，$\Gamma_1+\Gamma, \Gamma_2 \vdash a\{c/z\}:^{q} A$。于是，该 case 表达式由规则 ST-SubR 得出。

□

Theo注 B.6（保持性（定理 3.6））。若 $\Gamma \vdash a:\ ^{q} A$ 且 $\vdash a \rightsquigarrow a'$，则 $\Gamma \vdash a':\ ^{q} A$.

证明。通过对 $\Gamma \vdash a:^{q} A$ 进行归纳，并对 $\vdash a \rightsquigarrow a'$ 进行反演。

- 规则 ST-应用。已知：$\Gamma_1+\Gamma_2 \vdash b\ a^{r}:^{q} B$，其中 $\Gamma_1 \vdash b:^{q}\ ^{r} A \rightarrow B$ 且 $\Gamma_2 \vdash a:^{q \cdot r} A$。令 $\vdash b\ a^{r} \rightsquigarrow c$。根据反演：

– $\vdash b\ a^{r} \rightsquigarrow b'\ a^{r}$，其中 $\vdash b \rightsquigarrow b'$。需证明：$\Gamma_1 + \Gamma_2 \vdash b'\ a^{r} : {}^{q}B$。由归纳假设和规则 ST-App 可得。– $b = \lambda^{r} x : A'.b'$ 且 $\vdash b\ a^{r} \rightsquigarrow b'\{a/x\}$。需证明：$\Gamma_1 + \Gamma_2 \vdash b'\{a/x\} : {}^{q}B$。对 $\Gamma_1 \vdash \lambda^{r} x : A'.b' : {}^{q\ r}A \to B$ 进行反演，得到 $A' = A$ 和 $\Gamma_1, x : {}^{q_0 \cdot r}A \vdash b' : {}^{q_0}B$，其中某个 $q_0 <: q$。现在需要考虑两种情况。

  * $q_0 = 0$。由于 $q_0 <: q$，因此 $q = 0$。然后，根据替换引理，$\Gamma_1 + \Gamma_2 \vdash b\{a/x\} : {}^{0}B$。
  * $q_0 \neq 0$。根据引理 B.2，$\exists \Gamma'1$ 和 r'，使得 $\Gamma'1,\ x : {}^{r'}A \vdash b' : {}^{1}B$，且 $\Gamma_1 <: q0 \cdot \Gamma'1$，并且 $q0 \cdot r <: q0 \cdot r'$。由于 $q_0 \neq 0$，因此 $r <: r'$。于是，根据规则 ST-SubL，$\Gamma'1,\ x : {}^{r}A \vdash b' : {}^{1}B$。现在，根据引理 B.1，$q \cdot \Gamma'1,\ x : {}^{q \cdot r}A \vdash b' : {}^{q}B$。根据规则 ST-SubL，$\Gamma_1,\ x : {}^{q \cdot r}A \vdash b' : {}^{q}B$。然后，这个 case 表达式由替换引理得证。

- 规则 ST-LetPair。已知：$\Gamma_1 + \Gamma_2 \vdash 令_{q_0}(x^{r},\ y)$ 为 a，令：${}^{q}B$ 其中 $\Gamma_1 \vdash a : {}^{q_0\ r}A_1 \times A_2$ 且 $\Gamma_2,\ x : {}^{q \cdot q_0 \cdot r}A_1, y : {}^{q \cdot q_0}A_2 \vdash b : {}^{q}B$。令 $\vdash 令_{q_0}(x^{r},\ y)$ 为 a，令 $b \rightsquigarrow c$。通过反演：– $\vdash 令_{q_0}(x^{r},\ y)$ 为 a，令 $b \rightsquigarrow 令_{q_0}(x^{r},\ y)$ 为 a'，令 b，其中 $\vdash a \rightsquigarrow a'$。需要证明：$\Gamma_1 + \Gamma_2 \vdash 令_{q_0}(x^{r},\ y)$ 为 a'，令 $b : {}^{q}B$。由归纳假设和规则 ST-LetPair 可得。– $\vdash 令_{q_0}(x^{r},\ y)$ 为 $(a_1{}^{r},\ a_2)$，令 $b \rightsquigarrow b\{a_1/x\}\{a_2/y\}$。需要证明：$\Gamma_1 + \Gamma_2 \vdash b\{a_1/x\}\{a_2/y\} : {}^{q}B$。对 $\Gamma_1 \vdash (a_1{}^{r},\ a_2) : {}^{q \cdot q_0\ r}A_1 \times A_2$，进行反演可得：$\exists \Gamma_{11}, \Gamma_{12}$ 满足 $\Gamma_{11} \vdash a_1 : {}^{q \cdot q_0 \cdot r}A_1$ 且 $\Gamma_{12} \vdash a_2 : {}^{q \cdot q_0}A_2$ 和 $\Gamma_1 = \Gamma_{11} + \Gamma_{12}$。因此，此 case 表达式可通过两次应用替换引理得出。

- 规则 ST-LetUnit。已知：$\Gamma_1 + \Gamma_2 \vdash 令_{q_0}$ unit 为 a 代入 $b : {}^{q}B$ 其中 $\Gamma_1 \vdash a : {}^{q_0}$ 单元类型且 $\Gamma_2 \vdash b : {}^{q}B$。令 $\vdash let_{q_0}$ unit be a in b $\rightsquigarrow$ c。由反演可知：– $\vdash let_{q_0}$ unit be a in b $\rightsquigarrow let_{q_0}$ unit be a′ in b，当 $\vdash a \rightsquigarrow a'$。需要证明：$\Gamma_1 + \Gamma_2 \vdash let_{q_0}$ unit be a′ in b $: {}^{q}B$。由归纳假设和规则 ST-LetUnit 得出。– $\vdash let_{q_0}$ unit be unit in b $\rightsquigarrow$ b。需要证明：$\Gamma_1 + \Gamma_2 \vdash b : {}^{q}B$。在 $\mathbb{N}_{=}$ 的情况下，我们有，$\Gamma_1 = 0$。因此，这种情况可直接由前提 $\Gamma_2 \vdash b : {}^{q}B$ 得出。在 $\mathbb{N}_{\geq}$ 的情况下，该情况由规则 ST-SubL 得出。

— –

- 规则 ST-Case。有：$\Gamma_1 + \Gamma_2 \vdash case表达式_{q_0}\ a\ of\ x1.b1; x2.b2 : {}^{q}B$，其中 $\Gamma_1 \vdash a : {}^{q \cdot q_0}A_1 + A2$ 且 $\Gamma_2, x1 : {}^{q \cdot q_0}A_1 \vdash b1 : {}^{q}B$ 以及 $\Gamma_2, x2 : {}^{q \cdot q_0}A_2 \vdash b2 : {}^{q}B$。令 $\vdash case表达式_{q_0}\ a\ of\ x1.b1; x2.b_2 \rightsquigarrow c$。通过反演：– $\vdash case\ 表达式_{q_0}\ a\ of\ x1.b1;\ x2.b_2 \rightsquigarrow case表达式_{q_0}\ a'\ of\ x1.b1;\ x2.b2$，当 $\vdash a \rightsquigarrow a'$ 时。需证明：$\Gamma_1 + \Gamma_2 \vdash case\ 表达式_{q_0}\ a' of\ x1.b1\ ;\ x2.b2 : {}^{q}B$。根据归纳假设和规则 ST-Case 得证。– $\vdash case\ 表达式_{q_0}\ (inj1a1)\ of\ x1.b1; x2.b_2 \rightsquigarrow b1\{a1/x1\}$。需证明 $\Gamma_1 + \Gamma_2 \vdash b1\{a1/x1\} : {}^{q}B$。对 $\Gamma_1 \vdash inj1a1 : {}^{q \cdot q_0}A_1 + A2$ 进行反演，我们有，$\Gamma_1 \vdash a1 : {}^{q \cdot q_0}A1$。因此，该情形可通过应用替换引理得出。

– $\vdash \text{case}_{q_0}\ (\text{inj}_2 a_2)$ 的分支为 $x_1.b_1$ ； $x_2.b_2 \rightsquigarrow b_2\{a_2/x_2\}$。与前一种情况类似。

- 规则 ST-SubL 和ST-SubR。由归纳假设得出。

□

定理 B.7（进展（定理 3.7））。如果 $\emptyset \vdash a :^q A$，那么要么 a 是一个值，要么存在 a'使得 $\vdash a \rightsquigarrow a'$。

证明。对 $\emptyset \vdash a :^q A$ 作归纳。

- 规则 ST-Var。不适用，因为此处的上下文为空。- 规则 ST-App。已知： $\emptyset \vdash b\ a^r :^q B$，其中 $\emptyset \vdash b :^{q\ r} A \to B$，且 $\emptyset \vdash a :^{q \cdot r} A$。需要证明： $\exists c, \vdash b\ a^r \rightsquigarrow c$。由归纳假设，b 要么是一个值，要么是 $\vdash b \rightsquigarrow b'$。若 b 是一个值，则 $b = \lambda^r x : A.b'$，其中某个 b'。因此， $\vdash b\ a^r \rightsquigarrow b'\{a/x\}$。否则， $\vdash b\ a^r \rightsquigarrow b'\ a^r$。- 规则 ST-LetPair。已知： $\emptyset \vdash \text{let}_{q_0}(x^r, y)$ be a in b $:^q B$，其中 $\emptyset \vdash a :^{q \cdot q_0\ r} A_1 \times A2$，且 $x :^{q \cdot_{q_0} r} A1$， $y :^{q \cdot q_0} A_2 \vdash b :^q B$。需要证明： $\exists c, \vdash \text{let}_{q_0}(x^r, y)$ be a in b $\rightsquigarrow c$。由归纳假设，a 要么是一个值，要么是 $\vdash a \rightsquigarrow a'$。若 a 是一个值，则 $a = (a_1^r, a2)$。因此， $\vdash \text{let}_{q_0}(x^r, y)$ be a in b $\rightsquigarrow b\{a1/x\}\{a2/y\}$。否则， $\vdash \text{let}_{q_0}(x^r, y)$ be a in b $\rightsquigarrow \text{let}_{q_0}(x^r, y)$ be a′ in b。- 规则 ST-LetUnit。已知： $\emptyset \vdash \text{let}_{q_0}$ unit be a in b $:^q B$，其中 $\emptyset \vdash a :^{q \cdot q_0}$ 单元类型，且 $\emptyset \vdash b :^q B$。需要证明： $\exists c, \vdash \text{let}_{q_0}$ unit be a in b $\rightsquigarrow c$。由归纳假设，a 要么是一个值，要么是 $\vdash a \rightsquigarrow a'$。若 a 是一个值，则 a = unit。因此， $\vdash \text{let}_{q_0}$ unit be a in b $\rightsquigarrow b$。否则， $\vdash \text{let}_{q_0}$ unit be a in b $\rightsquigarrow \text{let}_{q_0}$ unit be a′ in b。- 规则 ST-Case。已知： $\emptyset \vdash \text{case}_{q_0}$ a 的分支为 x1.b1；x2.b2 $:^q B$，其中 $\emptyset \vdash a :^{q \cdot q_0} A_1 + A2$，且 $x1 :^{q \cdot q_0} A_1 \vdash b1 :^q B$， $x2 :^{q \cdot q_0} A_2 \vdash b2 :^q B$。需要证明： $\exists c, \vdash \text{case}_{q_0}$ a 的分支为 x1.b1；$x2.b_2 \rightsquigarrow c$。由归纳假设，a 要么是一个值，要么是 $\vdash a \rightsquigarrow a'$。若 a 是一个值，则 a = inj1 a1 或 a = inj2 a2。于是， $\vdash \text{case}_{q_0}$ a 的分支为 x1.b1；$x2.b_2 \rightsquigarrow b1\{a1/x1\}$，或者 $\vdash \text{case}_{q_0}$ a 的分支为 x1.b1；$x2.b_2 \rightsquigarrow b2\{a2/x2\}$。否则， $\vdash \text{case}_{q_0}$ a 的分支为 x1.b1；$x2.b_2 \rightsquigarrow \text{case}_{q_0}$ a′ 的分支为 x1.b1；x2.b2。

- 规则 ST-SubL 和 ST-SubR。根据归纳假设得出。
- 规则 ST-Lam、 ST-Pair和 ST-Unit。由这些规则赋予类型的项是值。

□

C 在简单类型 LDC 中的依赖性分析

引理 C.1（乘法（引理 3.8））。若 $\Gamma \vdash a :^\ell A$，则$m_0 \sqcup \Gamma \vdash a :^{m_0 \sqcup \ell} A$。

证明。对 $\Gamma \vdash a :^\ell A$ 进行归纳。

- 规则 ST-VarD。已知： $\top \sqcup \Gamma_1,\ x :^\ell A\ \top \sqcup \Gamma_2 \vdash x :^\ell A$。需要证明： $\top \sqcup \Gamma_1,\ x :^{m_0 \sqcup \ell} A\ \top \sqcup \Gamma_2 \vdash x :^{m_0 \sqcup \ell} A$。此情形可由规则 ST-VarD 得证。
- 规则 ST-LamD。已知： $\Gamma \vdash \lambda^m x : A.b :^{\ell\ m} A \to B$，其中 Γ， $x :^{\ell \sqcup m} A \vdash b :^\ell B$。需要证明： $m_0 \sqcup \Gamma \vdash \lambda^m x : A.b :^{m_0 \sqcup \ell\ m} A \to B$。根据归纳假设， $m_0 \sqcup \Gamma$， $x :^{m_0 \sqcup (\ell \sqcup m)} A \vdash b :^{m_0 \sqcup \ell} B$。因此，本 case 可通过规则 ST-LamD，利用 $\sqcup$ 的结合律得出。

• ST-AppD规则。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash ba^{m} :^{\ell} B$，其中 $\Gamma_1 \vdash b :^{\ell} {}^{m}A \to B$ 且 $\Gamma_2 \vdash a :^{\ell \sqcup m} A$。需要证明：$m_0 \sqcup (\Gamma_1 \sqcap \Gamma_2) \vdash ba^{m} :^{m_0 \sqcup \ell} B$。根据归纳假设，$m_0 \sqcup \Gamma_1 \vdash b :^{m_0 \sqcup \ell} {}^{m}A \to B$ 且 $m_0 \sqcup \Gamma_2 \vdash a :^{m_0 \sqcup (\ell \sqcup m)} A$。根据 ST-AppD 规则，利用 $\sqcup$ 的结合律，$(m_0 \sqcup \Gamma_1) \sqcap (m_0 \sqcup \Gamma_2) \vdash ba^{m} :^{m_0 \sqcup \ell} B$。现在，对于任意格中的元素 ℓ_1、ℓ_2 和 ℓ_3，$\ell_1 \sqcup (\ell_2 \sqcap \ell_3) \sqsubseteq (\ell_1 \sqcup \ell_2) \sqcap (\ell_1 \sqcup \ell_3)$。因此，此case 表达式根据 ST-SubLD 规则，利用上述关系即可得出。

• ST-PairD 规则。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash (a_1^{m}, a2) :^{\ell} {}^{m}A_1 \times A2$，其中 $\Gamma_1 \vdash a1 :^{\ell \sqcup m} A1$，且 $\Gamma_2 \vdash a2 :^{\ell} A2$。需要证明：$m_0 \sqcup (\Gamma_1 \sqcap \Gamma_2) \vdash (a_1^{m}, a2) :^{m_0 \sqcup \ell} {}^{m}A_1 \times A2$。根据归纳假设，$m_0 \sqcup \Gamma_1 \vdash a1 :^{m_0 \sqcup (\ell \sqcup m)} A1$，且 $m_0 \sqcup \Gamma_2 \vdash a2 :^{m_0 \sqcup \ell} A2$。根据 ST-PairD 规则，利用 $\sqcup$ 的结合律，$(m_0 \sqcup \Gamma_1) \sqcap (m_0 \sqcup \Gamma_2) \vdash (a_1^{m}, a2) :^{m_0 \sqcup \ell} {}^{m}A_1 \times A2$。对于 ℓ_1、ℓ_2、ℓ_3，我们有，$\ell_1 \sqcup (\ell_2 \sqcap \ell_3) \sqsubseteq (\ell_1 \sqcup \ell_2) \sqcap (\ell_1 \sqcup \ell_3)$。根据 ST-SubLD 规则，利用上述关系即可得到此情形。

• ST-LetPairD 规则。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{let } (x^{m}, y) \text{ be a in b} :^{\ell} B$，其中 $\Gamma_1 \vdash a :^{\ell} {}^{m}A_1 \times A2$ 且 $\Gamma_2, x :^{\ell \sqcup m} A1, y :^{\ell} A_2 \vdash b :^{\ell} B$。需证明：$m_0 \sqcup (\Gamma_1 \sqcap \Gamma_2) \vdash \text{let } (x^{m}, y) \text{ be a in b} :^{m_0 \sqcup \ell} B$。根据归纳假设，$m_0 \sqcup \Gamma_1 \vdash a :^{m_0 \sqcup \ell} {}^{m}A_1 \times A2$ 且 $m_0 \sqcup \Gamma_2, x :^{m_0 \sqcup (\ell \sqcup m)} A1, y :^{m_0 \sqcup \ell} A_2 \vdash b :^{m_0 \sqcup \ell} B$。此情形可由 ST-LetPairD 和 ST-SubLD 规则得出，利用 $\sqcup$ 的结合律及分配不等式。

• 规则 ST-UnitD。已知： $\top \sqcup \Gamma \vdash \text{unit} :^{\ell}$ 单元类型。
需证明： $\top \sqcup \Gamma \vdash \text{unit} :^{m_0 \sqcup \ell}$ 单元类型。由规则 ST-UnitD 得出。

• 规则 ST-LetUnitD。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{let unit be a in b} :^{\ell} B$，其中 $\Gamma_1 \vdash a :^{\ell}$ 单元类型，且 $\Gamma_2 \vdash b :^{\ell} B$。需证明：$m_0 \sqcup (\Gamma_1 \sqcap \Gamma_2) \vdash \text{let unit be a in b} :^{m_0 \sqcup \ell} B$。由归纳假设，$m_0 \sqcup \Gamma_1 \vdash a :^{m_0 \sqcup \ell}$ 单元类型，且 $m_0 \sqcup \Gamma_2 \vdash b :^{m_0 \sqcup \ell} B$。根据分配不等式，应用规则 ST-LetUnitD 和 ST-SubLD，即可得证。

• 规则 ST-Inj1D。已知： $\Gamma \vdash \text{inj1 a1} :^{\ell} A_1 + A2$，其中 $\Gamma \vdash a1 :^{\ell} A1$。
需要证明：$m_0 \sqcup \Gamma \vdash \text{inj1 a1} :^{m_0 \sqcup \ell} A_1 + A2$。根据归纳假设，$m_0 \sqcup \Gamma \vdash a1 :^{m_0 \sqcup \ell} A1$。因此，这个 case 表达式由规则 ST-Inj1D 得证。

• 规则 ST-Inj2D。类似于规则 ST-Inj1D。

• 规则 ST-CaseD。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{case a of x1.b1 ; x2.b2} :^{\ell} B$，其中 $\Gamma_1 \vdash a :^{\ell} A_1 + A2$，且 $\Gamma_2, x1 :^{\ell} A_1 \vdash b1 :^{\ell} B$，且 $\Gamma_2, x2 :^{\ell} A_2 \vdash b2 :^{\ell} B$。需要证明：$m_0 \sqcup (\Gamma_1 \sqcap \Gamma_2) \vdash \text{case a of x1.b1 ; x2.b2} :^{m_0 \sqcup \ell} B$。根据归纳假设，$m_0 \sqcup \Gamma_1 \vdash a1 :^{m_0 \sqcup \ell} A_1 + A2$，且 $m_0 \sqcup \Gamma_2, x1 :^{m_0 \sqcup \ell} A_1 \vdash b1 :^{m_0 \sqcup \ell} B$，且 $m_0 \sqcup \Gamma_2, x2 :^{m_0 \sqcup \ell} A_2 \vdash b2 :^{m_0 \sqcup \ell} B$。由规则 ST-CaseD 和 ST-SubLD，结合分配不等式，可得此情形。

• ST-SubLD 规则。已知： $\Gamma \vdash a :^{\ell} A$，其中 $\Gamma' \vdash a :^{\ell} A$，且 $\Gamma \sqsubseteq \Gamma'$。需要证明：$m_0 \sqcup \Gamma \vdash a :^{m_0 \sqcup \ell} A$。根据归纳假设，$m_0 \sqcup \Gamma' \vdash a :^{m_0 \sqcup \ell} A$。由于 $\Gamma \sqsubseteq \Gamma'$，因此 $m_0 \sqcup \Gamma \sqsubseteq m_0 \sqcup \Gamma'$。此 case 表达式由 ST-SubLD 规则得出。

• 规则 ST-SubRD。已知： $\Gamma \vdash a :^{\ell'} A$，其中 $\Gamma \vdash a :^{\ell} A$，且 $\ell \sqsubseteq \ell'$。
需要证明：$m_0 \sqcup \Gamma \vdash a :^{m_0 \sqcup \ell'} A$。根据归纳假设，$m_0 \sqcup \Gamma \vdash a :^{m_0 \sqcup \ell} A$。

由于 $\ell \sqsubseteq \ell'$，所以$m_0 \sqcup \ell \sqsubseteq m_0 \sqcup \ell'$。
此情形可由规则ST-SubRD得出。

□

引理 C.2（拆分（引理 3.9））。若 $\Gamma \vdash a : {}^{\ell_1 \sqcap \ell_2} A$，则存在 Γ_1 和 Γ_2，使得$\Gamma_1 \vdash a : {}^{\ell_1} A$ 且 $\Gamma_2 \vdash a : {}^{\ell_2} A$且 $\Gamma = \Gamma_1 \sqcap \Gamma_2$。

证明。有：$\Gamma \vdash a : {}^{\ell_1 \sqcap \ell_2} A$。根据规则 ST-SubRD，$\Gamma \vdash a : {}^{\ell_1} A$ 和 $\Gamma \vdash a : {}^{\ell_2} A$。通过令 $\Gamma_1 := \Gamma$ 和 $\Gamma_2 := \Gamma$，即可得证。□

引理 C.3（弱化（引理 3.10））。若 Γ_1，$\Gamma_2 \vdash a : {}^{\ell} A$，则 Γ_1，$z : {}^{\top} C$ $\Gamma_2 \vdash a : {}^{\ell} A$.

证明。对 Γ_1，$\Gamma_2 \vdash$ a :$^{\ell}$A 进行归纳。 □

引理 C.4（替换（引理 3.11））。若 Γ_1，z:^{m_0} C $\Gamma_2 \vdash a$:$^{\ell} A$ 且 $\Gamma \vdash c$:^{m_0} C且$\lfloor\Gamma_1\rfloor = \lfloor\Gamma\rfloor$，则 $\Gamma_1 \sqcap \Gamma$，$\Gamma_2 \vdash a\{c/z\}$:$^{\ell} A$。

证明。通过对 Γ_1，z :^{m_0}C $\Gamma_2 \vdash$ a :$^{\ell}$A 进行归纳。

- 规则ST-VarD。需要考虑三种情况。– $\top \sqcup \Gamma_{11}$，z :$^{\top}$C $\top \sqcup \Gamma_{12}$，x:$^{\ell}$A $\top \sqcup \Gamma_2 \vdash$x:$^{\ell}$A。此外，$\Gamma \vdash$c:$^{\top}$C，其中 $\lfloor\Gamma_{11}\rfloor = \lfloor\Gamma\rfloor$。需要证明：$\Gamma$，$\top \sqcup \Gamma_{12}$，x :$^{\ell}$A $\top \sqcup \Gamma_2 \vdash$x :$^{\ell}$A。由规则ST-VarD和ST-SubLD规则可得。– $\top \sqcup \Gamma_1$，x :$^{\ell}$A $\top \sqcup \Gamma_2 \vdash$x :$^{\ell}$A。此外，$\Gamma \vdash$a :$^{\ell}$A，其中 $\lfloor\Gamma_1\rfloor = \lfloor\Gamma\rfloor$。需要证明：$\Gamma$，$\top \sqcup \Gamma_2 \vdash$ a:$^{\ell}$A。由引理 C.3 可得。– $\top \sqcup \Gamma_1$，x:$^{\ell}$A $\top \sqcup \Gamma_{21}$，z:$^{\top}$C $\top \sqcup \Gamma_{22} \vdash$x:$^{\ell}$A。此外，$\Gamma_{31}$，x:^{m}A $\Gamma_{32} \vdash$c:$^{\top}$C，其中$\lfloor\Gamma_{31}\rfloor = \lfloor\Gamma_1\rfloor$ 和 $\lfloor\Gamma_{32}\rfloor = \lfloor\Gamma_{21}\rfloor$。需要证明：$\Gamma_{31}$，x:($\ell \sqcap m$) A Γ_{32}，$\top \sqcup \Gamma_{22} \vdash$x :$^{\ell}$A。由规则 ST-VarD 和 ST-SubLD 规则可得。

- 规则ST-LamD。已知：Γ_1,z :^{m_0}C $\Gamma_2 \vdash \lambda^{m}$x:A.b :$^{\ell}$ ^{m}A $\rightarrow$B，其中 Γ_1, z:^{m_0}C Γ_2, x:$^{\ell \sqcup m}$ A $\vdash$ b :$^{\ell}$B。另有，$\Gamma \vdash$ c :^{m_0} C，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_1\rfloor$。需证明：$\Gamma_1 \sqcap \Gamma, \Gamma_2 \vdash \lambda^{m}$x:A.b{c/z }: $^{\ell}$ ^{m}A $\rightarrow$B 。由归纳假设和规则 ST-LamD 可得。
- ST-AppD 规则。已知：$\Gamma_{11} \sqcap \Gamma_{12}$，z :$^{m_{01} \sqcap m_{02}}$C $\Gamma_{21} \sqcap \Gamma_{22} \vdash$b a^{m}:$^{\ell}$B，其中 Γ_{11}，z :$^{m_{01}}$C $\Gamma_{21} \vdash$b :$^{\ell}$ ^{m}A $\rightarrow$B，且 Γ_{12}，z :$^{m_{02}}$C $\Gamma_{22} \vdash$a :$^{\ell \sqcup m}$A。此外，$\Gamma \vdash$c :$^{m_{01} \sqcap m_{02}}$C，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_{11}\rfloor$。需要证明：$\Gamma_{11} \sqcap \Gamma_{12} \sqcap \Gamma, \Gamma_{21} \sqcap \Gamma_{22} \vdash$b{c/z } a{c/z }^{m}:$^{\ell}$B。根据引理 C.2，$\exists\Gamma_{31}$, Γ_{32} 使得 $\Gamma_{31} \vdash$ c :$^{m_{01}}$ C 且 $\Gamma_{32} \vdash$ c :$^{m_{02}}$ C，并且 $\Gamma = \Gamma_{31} \sqcap \Gamma_{32}$。根据归纳假设，$\Gamma_{11} \sqcap \Gamma_{31}$, $\Gamma_{21} \vdash$b{c/z } :$^{\ell}$ ^{m}A $\rightarrow$B，且 $\Gamma_{12} \sqcap \Gamma_{32}, \Gamma_{22} \vdash$a{c/z } :$^{\ell \sqcup m}$A。于是此情形可由 ST-AppD 规则得出。
- ST-PairD 规则。已知：$\Gamma_{11} \sqcap \Gamma_{12}$, z :$^{m_{01} \sqcap m_{02}}$C $\Gamma_{21} \sqcap \Gamma_{22} \vdash$(a$_1{}^{m}$, a2) :$^{\ell}$ ^{m}A$_1$ $\times$A2，其中Γ_{11}, z :$^{m_{01}}$C $\Gamma_{21} \vdash$a1 :$^{\ell \sqcup m}$A1 且 Γ_{12}, z :$^{m_{02}}$C $\Gamma_{22} \vdash$a2 :$^{\ell}$A2。此外，$\Gamma \vdash$c :$^{m_{01} \sqcap m_{02}}$C，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_{11}\rfloor$。需证明：$\Gamma_{11} \sqcap \Gamma_{12} \sqcap \Gamma, \Gamma_{21} \sqcap \Gamma_{22} \vdash$(a1{c/z }^{r}, a2{c/z }) :$^{\ell}$ ^{m}A$_1$ $\times$A2。由引理 C.2，$\exists\Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash$ c :$^{m_{01}}$ C 且 $\Gamma_{32} \vdash$ c :$^{m_{02}}$ C，并且 $\Gamma = \Gamma_{31} \sqcap \Gamma_{32}$。根据归纳假设，$\Gamma_{11} \sqcap \Gamma_{31}, \Gamma_{21} \vdash$a1{c/z } :$^{\ell \sqcup m}$A1 且 $\Gamma_{12} \sqcap \Gamma_{32}, \Gamma_{22} \vdash$a2{c/z } :$^{\ell}$A2。因此，这个 case 表达式可由 ST-PairD 规则推出。
- 规则 ST-LetPairD。有：$\Gamma_{11} \sqcap \Gamma_{12}$，z :$^{m_{01} \sqcap m_{02}}$C $\Gamma_{21} \sqcap \Gamma_{22} \vdash$令(x ^{m}, y) 为 a 在 b 中:$^{\ell}$B，其中 Γ_{11}, z :$^{m_{01}}$C $\Gamma_{21} \vdash$a :$^{\ell}$ ^{m}A$_1 \times$A$_2$ 以及 Γ_{12}，z :$^{m_{02}}$C Γ_{22}，x :$^{\ell \sqcup m}$A$_1$，y :$^{\ell}$A$_2 \vdash$b :$^{\ell}$B。此外，$\Gamma \vdash$c :$^{m_{01} \sqcap m_{02}}$ C，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_{11}\rfloor$。

需证明：$\Gamma_{11} \sqcap \Gamma_{12} \sqcap \Gamma, \Gamma_{21} \sqcap \Gamma_{22} \vdash$ 令 (x ^{m}, y)为 a{c/z} 中的b{c/z }:$^{\ell}$B。根据引理 C.2, $\exists \Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash$ c:$^{m_{01}}$ C且 $\Gamma_{32} \vdash$ c:$^{m_{02}}$ C 以及 $\Gamma = \Gamma_{31} \sqcap \Gamma_{32}$。根据归纳假设, $\Gamma_{11} \sqcap \Gamma_{31}, \Gamma_{21} \vdash$ a{c/z }:$^{\ell\ m}$A$_1 \times$A$_2$并且 $\Gamma_{12} \sqcap \Gamma_{32}, \Gamma_{22}$, x :$^{\ell \sqcup m}A_1$,y :$^{\ell}A_2 \vdash$ b{c/z } :$^{\ell}$B。因此，此 case 表达式由 ST-LetPairD 规则得出。

- 规则 ST-UnitD。有： $\top \sqcup \Gamma_1$, z: $^{\top} C$ $\top \sqcup \Gamma_2 \vdash$ 单元类型：$^{\ell}$单元类型。此外 $\Gamma \vdash c$: $^{\top} C$其中$\lfloor \Gamma \rfloor = \lfloor \Gamma_1 \rfloor$。需要证明： Γ, $\top \sqcup \Gamma_2 \vdash$ 单元类型：$^{\ell}$单元类型。由规则 ST-UnitD和规则 ST-SubLD得证。
- 规则 ST-LetUnitD。已知： $\Gamma_{11} \sqcap \Gamma_{12}$, z :$^{m_{01} \sqcap m_{02}}$C $\Gamma_{21} \sqcap \Gamma_{22} \vdash$let unit be a in b :$^{\ell}$B，其中 Γ_{11}, z :$^{m_{01}}$C $\Gamma_{21} \vdash$a :$^{\ell}$单元类型，且 Γ_{12}, z :$^{m_{02}}$C $\Gamma_{22} \vdash$b :$^{\ell}$B。此外， $\Gamma \vdash$c :$^{m_{01} \sqcap m_{02}}$C，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_{11} \rfloor$。需证明： $\Gamma_{11} \sqcap \Gamma_{12} \sqcap \Gamma, \Gamma_{21} \sqcap \Gamma_{22} \vdash$let unit be a{c/z } in b{c/z } :$^{\ell}$B。由引理 C.2, $\exists \Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash$ c :$^{m_{01}}$ C 且 $\Gamma_{32} \vdash$ c :$^{m_{02}}$ C，并且 $\Gamma = \Gamma_{31} \sqcap \Gamma_{32}$。由归纳假设, $\Gamma_{11} \sqcap \Gamma_{31}, \Gamma_{21} \vdash$a{c/z } :$^{\ell}$单元类型，且 $\Gamma_{12} \sqcap \Gamma_{32}, \Gamma_{22} \vdash$b{c/z } :$^{\ell}$B。因此，此情形可由规则 ST-LetUnitD 得出。
- 规则 ST-Inj1D。已知： Γ_1, z :^{m_0}C $\Gamma_2 \vdash$inj1 a1 :$^{\ell}$A$_1$ +A2，其中 Γ_1, z :^{m_0}C $\Gamma_2 \vdash$a1 :$^{\ell}$ A1。此外， $\Gamma \vdash$c :^{m_0} C，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_1 \rfloor$。需证明： $\Gamma_1 \sqcap \Gamma, \Gamma_2 \vdash$ inj1 a1{c/z } :$^{\ell}$A$_1$ + A2。根据归纳假设， $\Gamma_1 \sqcap \Gamma, \Gamma_2 \vdash$ a1{c/z } :$^{\ell}$A1。因此，这个 case 表达式可由规则 ST-Inj1D 得出。
- 规则ST-Inj2D。与规则ST-Inj1D类似。
- 规则 ST-CaseD。已知： $\Gamma_{11} \sqcap \Gamma_{12}$, z :$^{m_{01} \sqcap m_{02}}$C $\Gamma_{21} \sqcap \Gamma_{22} \vdash$case 表达式a of x1.b1; x2.b2 :$^{\ell}$B, 其中 Γ_{11}, z :$^{m_{01}}$C $\Gamma_{21} \vdash$a:$^{\ell}$A$_1$ +A2，且 Γ_{12},z :$^{m_{02}}$C Γ_{22}, x1 :$^{\ell}$A$_1 \vdash$b1 :$^{\ell}$B，以及Γ_{12},z:$^{m_{02}}$C Γ_{22}, x2:$^{\ell}$A$_2 \vdash$b2 :$^{\ell}$B。此外， $\Gamma \vdash$c :$^{m_{01} \sqcap m_{02}}$C，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_{11} \rfloor$。需要证明： $\Gamma_{11} \sqcap \Gamma_{12} \sqcap \Gamma, \Gamma_{21} \sqcap \Gamma_{22} \vdash$ case 表达式a{c/z }ofx1.b1{c/z};x2.b2{c/z} :$^{\ell}$B 。根据引理C.2， $\exists \Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash$ c :$^{m_{01}}$ C且 $\Gamma_{32} \vdash$ c:$^{m_{02}}$ C，并且 $\Gamma = \Gamma_{31} \sqcap \Gamma_{32}$。根据归纳假设， $\Gamma_{11} \sqcap \Gamma_{31}, \Gamma_{21} \vdash$ a{c/z }:$^{\ell}$A$_1$ +A2，且 $\Gamma_{12} \sqcap \Gamma_{32}$, Γ_{22}, x1:$^{\ell}A_1 \vdash$b1{c/z }:$^{\ell}$B，以及$\Gamma_{12} \sqcap \Gamma_{32}, \Gamma_{22}$, x2 :$^{\ell}A_2 \vdash$b2{c/z} :$^{\ell}$B。因此，本情况由规则 ST-CaseD 得出。
- ST-SubLD规则。已知： Γ_1, z :^{m_0}C $\Gamma_2 \vdash$a:$^{\ell}$A，其中 $\Gamma'1$, z :$^{m'_0}$C $\Gamma'_2 \vdash$a :$^{\ell}$A，其中 $\Gamma_1 \sqsubseteq \Gamma'1$且 $m_0 \sqsubseteq m'0$ 以及 $\Gamma_2 \sqsubseteq \Gamma'2$。此外， $\Gamma \vdash$ c :^{m_0} C，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_1 \rfloor$。需要证明： $\Gamma_1 \sqcap \Gamma, \Gamma_2 \vdash$ a{c/z } :$^{\ell}$A。由于 $m_0 \sqsubseteq m'0$，根据ST-SubRD规则， $\Gamma \vdash$ c :$^{m'_0}$ C。根据归纳假设， $\Gamma'_1 \sqcap \Gamma, \Gamma'_2 \vdash$ a{c/z }:$^{\ell}$A。因此，此 case 表达式由 ST-SubLD 规则得出。
- ST-SubRD 规则。已知： Γ_1, z :^{m_0}C $\Gamma_2 \vdash$a :$^{\ell'}$ A，其中 Γ_1, z :^{m_0}C $\Gamma_2 \vdash$a :$^{\ell}$A 且 $\ell \sqsubseteq \ell'$。此外， $\Gamma \vdash$c :^{m_0} C，其中 $\lfloor \Gamma \rfloor = \lfloor \Gamma_1 \rfloor$。需证明： $\Gamma_1 \sqcap \Gamma, \Gamma_2 \vdash$ a{c/z } :$^{\ell'}$ A。根据归纳假设, $\Gamma_1 \sqcap \Gamma, \Gamma_2 \vdash$ a{c/z } :$^{\ell}$A。因此，此 case 表达式由 ST-SubRD 规则即可推出。

□

引理 C.5（受限升级）。若 Γ_1, x :^{m}A $\Gamma_2 \vdash$b :$^{\ell}$B 且 $\ell_0 \sqsubseteq \ell$，则 Γ_1, x :$^{\ell_0 \sqcup m}$A $\Gamma_2 \vdash$b :$^{\ell}$B。

P顶。 根据引理 C.1， $\ell_0 \sqcup \Gamma_1$， $x : {}^{\ell_0 \sqcup m} A$ $\ell_0 \sqcup \Gamma_2 \vdash b : {}^{\ell_0 \sqcup \ell} B$.
Since $\ell_0 \sqsubseteq \ell$， $\ell_0 \sqcup \Gamma_1$， $x : {}^{\ell_0 \sqcup m} A$ $\ell_0 \sqcup \Gamma_2 \vdash b : {}^{\ell} B$。
By 规则 ST-SubLD， Γ_1， $x : {}^{\ell_0 \sqcup m} A$ $\Gamma_2 \vdash b : {}^{\ell} B$，因为 $\Gamma_1 \sqsubseteq \ell_0 \sqcup \Gamma_1$ and $\Gamma_2 \sqsubseteq \ell_0 \sqcup \Gamma_2$。□

定理 C.6（保持性（定理 3.12））。如果 $\Gamma \vdash a : {}^{\ell} A$ 且 $\vdash a \rightsquigarrow a'$，那么 $\Gamma \vdash a' : {}^{\ell} A$。

证明。对 $\Gamma \vdash a : {}^{\ell} A$ 做归纳，并对 $\vdash a \rightsquigarrow a'$ 做反演。

- 规则 ST-AppD。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash b\ a^m : {}^{\ell} B$，其中 $\Gamma_1 \vdash b : {}^{\ell\ m} A \to B$ 且 $\Gamma_2 \vdash a : {}^{\ell \sqcup m} A$。令 $\vdash b\ a^m \rightsquigarrow c$。通过反演： $- \vdash b\ a^m \rightsquigarrow b'\ a^m$，当 $\vdash b \rightsquigarrow b'$ 时。需要证明： $\Gamma_1 \sqcap \Gamma_2 \vdash b'\ a^m : {}^{\ell} B$。由归纳假设和规则 ST-AppD 得出。$- b = \lambda^m x : A'.b'$，且 $\vdash b\ a^m \rightsquigarrow b'\{a/x\}$。需要证明： $\Gamma_1 \sqcap \Gamma_2 \vdash b'\{a/x\} : {}^{\ell} B$。对 $\Gamma_1 \vdash \lambda^m x : A'.b' : {}^{\ell\ m} A \to B$ 做反演，我们得到 $A' = A$ 且 Γ_1， $x : {}^{\ell_0 \sqcup m} A \vdash b' : {}^{\ell_0} B$，其中某个 $\ell_0 \sqsubseteq \ell$。根据 ST-SubRD 规则， Γ_1， $x : {}^{\ell_0 \sqcup m} A \vdash b' : {}^{\ell} B$。根据引理 C.5， Γ_1， $x : {}^{\ell \sqcup m} A \vdash b' : {}^{\ell} B$。因此，本 case 由替换引理得出。

- 规则 ST-LetPairD。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{let } (x^m, y) \text{ be a in } b : {}^{\ell} B$，其中 $\Gamma_1 \vdash a : {}^{\ell\ m} A_1 \times A2$，且 $\Gamma_2\ x : {}^{\ell \sqcup m} A1$， $y : {}^{\ell} A_2 \vdash b : {}^{\ell} B$。令 $\vdash \text{let } (x^m, y) \text{ be a in } b \rightsquigarrow c$。通过反演： $- \vdash \text{let } (x^m, y) \text{ be a in } b \rightsquigarrow \text{let } (x^m, y) \text{ be a' in } b$，当 $\vdash a \rightsquigarrow a'$。需要证明： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{let } (x^m, y) \text{ be a' in } b : {}^{\ell} B$。由归纳假设和规则 ST-LetPairD 得出。$- \vdash \text{let } (x^m, y) \text{ be } (a_1{}^m, a2) \text{ in } b \rightsquigarrow b\{a1/x\}\{a2/y\}$。需要证明： $\Gamma_1 \sqcap \Gamma_2 \vdash b\{a1/x\}\{a2/y\} : {}^{\ell} B$。对 $\Gamma_1 \vdash (a_1{}^m, a2) : {}^{\ell\ m} A_1 \times A2$ 进行反演，我们有： $\exists \Gamma_{11}, \Gamma_{12}$ 使得 $\Gamma_{11} \vdash a1 : {}^{\ell \sqcup m} A1$ 且 $\Gamma_{12} \vdash a2 : {}^{\ell} A2$ 以及 $\Gamma_1 = \Gamma_{11} \sqcap \Gamma_{12}$。因此，这种情况可通过两次应用替换引理得出。

- 规则 ST-LetUnitD。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{let unit be a in } b : {}^{\ell} B$，其中 $\Gamma_1 \vdash a : {}^{\ell} \text{Unit}$ 且 $\Gamma_2 \vdash b : {}^{\ell} B$。令 $\vdash \text{let unit be a in } b \rightsquigarrow c$。根据反演： $- \vdash \text{let unit be a in } b \rightsquigarrow \text{let unit be a' in b}$，其中 $\vdash a \rightsquigarrow a'$。需要证明： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{let unit be a' in } b : {}^{\ell} B$。由归纳假设和规则 ST-LetUnitD 得证。$- \vdash \text{let unit be unit in } b \rightsquigarrow b$。需要证明： $\Gamma_1 \sqcap \Gamma_2 \vdash b : {}^{\ell} B$。由 ST-SubLD 规则得证。

- 规则 ST-CaseD。已知： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{case a of x1.b1 ; x2.b2} : {}^{\ell} B$，其中 $\Gamma_1 \vdash a : {}^{\ell} A_1 + A2$，且 Γ_2、$x1 : {}^{\ell} A_1 \vdash b1 : {}^{\ell} B$，以及 Γ_2、$x2 : {}^{\ell} A_2 \vdash b2 : {}^{\ell} B$。设 $\vdash \text{case a of x1.b1 ; x2.b}_2 \rightsquigarrow c$。通过反演： $- \vdash \text{case a of x1.b1 ; x2.b}_2 \rightsquigarrow \text{case a' of x1.b1 ; x2.b2}$，其中 $\vdash a \rightsquigarrow a'$。需要证明： $\Gamma_1 \sqcap \Gamma_2 \vdash \text{case a' of x1.b1 ; x2.b2} : {}^{\ell} B$。由归纳假设和规则 ST-CaseD 得出。$- \vdash \text{case (inj1 a1) of x1.b1 ; x2.b}_2 \rightsquigarrow b1\{a1/x1\}$。需要证明 $\Gamma_1 \sqcap \Gamma_2 \vdash b1\{a1/x1\} : {}^{\ell} B$。对 $\Gamma_1 \vdash \text{inj1 a1} : {}^{\ell} A_1 + A2$ 进行反演，可得： $\Gamma_1 \vdash a1 : {}^{\ell} A1$。因此，此情形可通过应用替换引理得出。

– $\vdash$ case (inj$_2$ a$_2$) of x$_1$.b$_1$; x$_2$.b$_2$ $\rightsquigarrow$ b$_2${a$_2$/x$_2$}。
与前一种情况类似。

- 规则 ST-SubLD 和 ST-SubRD。由归纳假设可得。

□

定理 C.7（进展（定理 3.13））。若 $\emptyset \vdash a :^{\ell} A$，则 a 要么是一个值，要么存在a 是值，或者存在a' 使得 $\vdash a \rightsquigarrow a'$。

证明。对 $\emptyset \vdash$ a $:^{\ell}$ A 进行归纳。

- 规则 ST-VarD。此处上下文为空，因此该规则不适用。
- 规则 ST-AppD。已知：$\emptyset \vdash$ b a$^m :^{\ell}$ B，其中 $\emptyset \vdash$ b $:^{\ell}{}^m$A $\rightarrow$ B 且 $\emptyset \vdash$ a : $^{\ell \sqcup m}$A。需要证明：$\exists c, \vdash$ b a$^m \rightsquigarrow$ c。根据归纳假设，b 要么是一个值，要么是 $\vdash$ b $\rightsquigarrow$ b′。若 b 是一个值，则 b $= \lambda^m$x : A.b′ 对某个 b′ 成立。因此，$\vdash$ b a$^m \rightsquigarrow$ b′{a/x }。否则，$\vdash$ b a$^m \rightsquigarrow$ b′ a^m。
- ST-LetPairD 规则。有：$\emptyset \vdash$ let (x^m,y)beain b $:^{\ell}$ B，其中 $\emptyset \vdash$ a:$^{\ell}{}^m$A$_1$ $\times$A2 且x: $^{\ell \sqcup m}$A1，y :$^{\ell}$A$_2 \vdash$ b :$^{\ell}$B。需证明：$\exists c, \vdash$ let(x^m,y) be a inb $\rightsquigarrow$ c。根据归纳假设，a 要么是值，要么 $\vdash$ a $\rightsquigarrow$ a′。如果a 是值，则 a =(a$_1{}^m$, a2)。因此，$\vdash$let(x ^{m}, y) bea in b $\rightsquigarrow$ b{a1/x}{a2/y }。否则，$\vdash$ let (x^m, y) be a in b $\rightsquigarrow$ let (x^m, y) be a′ in b。

- 规则 ST-LetUnitD。已知：$\emptyset \vdash$ 在 $^{\ell}$B 中，令 unit 为 a，结果为 b：$\emptyset \vdash$ a :$^{\ell}$Unit 且 $\emptyset \vdash$ b :$^{\ell}$B。需要证明：$\exists c, \vdash$ 令 unit 为 a，结果为 b $\rightsquigarrow$ c。根据归纳假设，a 要么是一个值，要么 $\vdash$ a $\rightsquigarrow$ a′。如果 a 是一个值，则 a $=$ unit。因此，$\vdash$ 令 unit 为 a，结果为 b $\rightsquigarrow$ b。否则，$\vdash$ 令 unit 为 a，结果为 b $\rightsquigarrow$ 令 unit 为 a′，结果为 b。
- 规则 ST-CaseD。已知：$\emptyset \vdash$ case a of x1.b1 ; x2.b2 :$^{\ell}$B，其中 $\emptyset \vdash$ a :$^{\ell}$A$_1$ + A2，且 x1 :$^{\ell}$A$_1 \vdash$b1 :$^{\ell}$B，x2 :$^{\ell}$A$_2 \vdash$ b2 :$^{\ell}$B。需证明：$\exists c, \vdash$ case a of x1.b1 ; x2.b$_2 \rightsquigarrow$ c。由归纳假设，a 要么是一个值，要么满足 $\vdash$ a $\rightsquigarrow$ a′。若 a 是一个值，则 a $=$ inj1 a1 或 a $=$ inj2 a2。于是，$\vdash$ case a of x1.b1 ; x2.b$_2 \rightsquigarrow$ b1{a1/x1}，或 $\vdash$ case a of x1.b1 ; x2.b$_2 \rightsquigarrow$ b2{a2/x2}。否则，$\vdash$ case a of x1.b1 ; x2.b$_2 \rightsquigarrow$ case a′ of x1.b1 ; x2.b2。

- 规则 ST-SubLD 和 ST-SubRD。由归纳假设可得。
- 规则ST-LamD、ST-PairD和ST-UnitD。由这些规则进行类型判断的项是值。

□

D 面向简单类型LDC的堆语义

引理 D.1（相似性）。如果 $[H]a \Longrightarrow_S^q [H']a'$，那么 $a\{H\} = a'\{H'\}$，或者 $\vdash a\{H\} \rightsquigarrow a'\{H'\}$。这里，$a\{H\}$ 表示将a 中的定义按H中的逆序代入后得到的项。

证明。根据 [H]a $\Longrightarrow_S^q$ [H ′]a′ 进行归纳。 □

引理 D.2（未改变（引理 4.1））。如果 [H$_1$,x $\overset{r}{\mapsto}$ a，H$_2$]c $\Longrightarrow_S^q$ [H ′$_1$，x $\overset{r'}{\mapsto}$ a，H′$_2$]c′（其中$|H_1| = |H'_1|$）且 $\neg(\exists q_0,\ r = q + q_0)$，则 r ′ $= r$。

证明。对 [H$_1$,x $\overset{r}{\mapsto}$ a，H$_2$]c $\Longrightarrow_S^q$ [H ′$_1$，x $\overset{r'}{\mapsto}$ a，H ′$_2$]c′ 进行归纳。有两个值得关注的 case 表达式：

- 规则 堆步进-变量。有：$[H_1, x \overset{r}{\mapsto} a, H_2]c \Longrightarrow_S^q [H'_1, x \overset{r'}{\mapsto} a, H'_2]c'$，其中 c 是一个变量它是在 H_1 或 H_2 中定义的。无论哪种情况，$r' = r$。注意，c 不可能是 x，否则由于条件 $\neg(\exists q_0, r = q + q_0)$，此规则将不适用。
- 规则 HeapStep-Discard。我们根据参数化结构进行 case 分析：– 当参数化结构为 $\mathbb{N}_=$ 时，此规则是多余的。– 当参数化结构为 $\mathbb{N}_\geq$ 时，我们有：$[H1, x \overset{r}{\mapsto} a, H_2]c \Longrightarrow_S^{q'} [H'1, x \overset{r'}{\mapsto} a, H'_2]c'$，其中 $[H1, x \overset{r}{\mapsto} a, H_2]c \Longrightarrow_S^q [H'1, x \overset{r'}{\mapsto} a, H'_2]c'$，且 $q <: q'$。此外，$\neg(\exists q_0, r = q' + q_0)$。现在，由于 $q <: q'$，我们知道，$\exists q1, q = q' + q1$。于是，若存在某个 q2，使等式 $r = q + q2$ 成立，就会导致矛盾，因为在这种情况下，等式 $r = q' + (q_1 + q2)$ 也成立。因此，$\neg(\exists q_0, r = q + q_0)$。于是，此情形由归纳假设得证。– 当参数化结构为格 L 时，我们有：$[H1, x \overset{m}{\mapsto} a, H_2]c \Longrightarrow_S^{\ell'} [H'1, x \overset{m'}{\mapsto} a, H'_2]c'$，其中 $[H1, x \overset{m}{\mapsto} a, H_2]c \Longrightarrow_S^{\ell} [H'1, x \overset{m'}{\mapsto} a, H'_2]c'$，且 $\ell \sqsubseteq \ell'$。此外，$\neg(m \sqsubseteq \ell')$。那么，$\neg(m \sqsubseteq \ell)$，因为否则 $m \sqsubseteq \ell \sqsubseteq \ell'$。于是，此情形由归纳假设得证。

□

引理 D.3（无关性（引理 4.2））。如果 $[H_1, x \overset{r}{\mapsto} a, H_2]c \Longrightarrow_{S \cup fv\ b}^q [H'1, x \overset{r'}{\mapsto} a, H'_2]c'$（其中 $|H_1| = |H'_1|$）且 $\neg(\exists q_0, r = q + q_0)$，那么 $[H_1, x \overset{r}{\mapsto} b, H_2]c \Longrightarrow_{S \cup fv\ a}^q [H'1, x \overset{r'}{\mapsto} b, H'_2]c'$。

证明。对 $[H_1, x \overset{r}{\mapsto} a, H_2]c \Longrightarrow_{S \cup fv\ b}^q [H'_1, x \overset{r'}{\mapsto} a, H'_2]c'$ 进行归纳。 □

引理 D.4. 若 $H \models \Gamma_1 + \Gamma_2$ 且 $\Gamma_2 \vdash a :^q A$ 且 $q \neq 0$，则 a 要么是一个值，要么存在H'，Γ'_2, a' 使得：

- $[H]a \Longrightarrow_S^q [H']a'$
- $H' \models \Gamma_1 + \Gamma'_2$
- $\Gamma'_2 \vdash a' :^q A$

（注意，+ 在此处被重载：$\Gamma_1 + \Gamma_2$ 表示上下文 Γ_1 和 Γ_2 的加法，在必要时先对它们进行填充。）

证明。通过对 $\Gamma_2 \vdash a :^q A$ 进行归纳。

- 规则 ST-Var。具有$0 \cdot \Gamma_{21}$、$x :^{q_2} A$、$0 \cdot \Gamma_{22} \vdash x :^{q_2} A$。此外，$H \models (\Gamma_{11}, x :^{q_1} A\ \Gamma_{12}) + (0 \cdot \Gamma_{21}, x :^{q_2} A, 0 \cdot \Gamma_{22})$。通过反演，∃H1、q、a、H2、$\Gamma_0$ 以及 q0，使得 $H = H1, x \overset{q}{\mapsto} a$、H2且 $\Gamma_0 \vdash a :^q A$以及$q = q_1 + q_2 + q0$。根据引理B.3，$\exists \Gamma_{01}, \Gamma_{02}$ 使得 $\Gamma_{01} \vdash a :^{q_0+q_1} A$且 $\Gamma_{02} \vdash a :^{q_2} A$ 以及 $\Gamma_0 = \Gamma_{01} + \Gamma_{02}$。那么，我们有，– $[H1, x \overset{q_0+q_1+q_2}{\mapsto} a, H_2]x \Longrightarrow_S^{q_2} [H1, x \overset{q_0+q_1}{\mapsto} a, H_2]a$。– $H1, x \overset{q_0+q_1}{\mapsto} a, H_2 \models (\Gamma_{11}, x :^{q_1} A\ \Gamma_{12}) + (\Gamma_{02}, x :^0 A, 0 \cdot \Gamma_{22})$。– $\Gamma_{02}, x :^0 A, 0 \cdot \Gamma_{22} \vdash a :^{q_2} A$。

- 规则 ST-应用。已知：$\Gamma_{21} + \Gamma_{22} \vdash b\ a^r :^q B$ 其中 $\Gamma_{21} \vdash b :^{q\ r} A \to B$ 且 $\Gamma_{22} \vdash a :^{q \cdot r} A$。进一步，$H \models \Gamma_1 + (\Gamma_{21} + \Gamma_{22})$。根据归纳假设，如果$b$步，则 $[H]b \Longrightarrow_{S \cup fv\ a}^q [H']b'$并且 $\Gamma'_{21} \vdash b' :^{q\ r} A \to B$并且$H' \models$

$\Gamma_1 + (\Gamma'_{21} + \Gamma_{22})$。然后，$[H]b\ a^r \Longrightarrow^q_S [H']b'\ a^r$ 并且 $\Gamma'_{21} + \Gamma_{22} \vdash b'\ a^r : ^q B$ 并且 $H' \models \Gamma_1 + (\Gamma'_{21} + \Gamma_{22})$。否则，b 是一个值。通过反演 $b = \lambda^r x : A.b'$。此外，通过反演，$\Gamma_{21}, x :^{q\cdot r} A \vdash b' :^q B$。然后，$[H](\lambda^r x : A.b')\ a \Longrightarrow^q_S [H, x \overset{q\ r}{\mapsto} \cdot\ a]b'$（假设 x 是新鲜的）。进一步，$H, x \overset{q\ r}{\mapsto} \cdot\ a \models (\Gamma_1 + \Gamma_{21}), x :^{q\cdot r} A$。• 规则 ST-LetUnit。可得：$\Gamma_{21} + \Gamma_{22} \vdash \text{let}^{q_0}\ \text{unit be a in b} :^q B$，其中 $\Gamma_{21} \vdash a :^{q\cdot q_0}$ 单元类型 and$\Gamma_{22} \vdash b :^q B$。进一步，$H \models \Gamma_1 + (\Gamma_{21} + \Gamma_{22})$。

根据归纳假设，如果 a 进行一步求值，那么 $[H]a \Longrightarrow^q \cdot \ {}^{q_0}_{S \cup}$ 自由变量 $b\ [H',]a'$ 且 $H', \models \Gamma_1 + (\Gamma'_{21} + \Gamma_{22})$ 且 $\Gamma'_{21} \vdash a' :^q \cdot$

q_0 单元类型。

那么，$[H]\text{let}^{q_0}\ \text{unit be a in b} \Longrightarrow^q \cdot \ {}_S{}^{q_0}$ $[H']$令$_{q_0}$ 单元类型为′，则为 b。

根据规则 HeapStep-Discard，$[H]\text{let}^{q_0}\ \text{unitbe a inb} \Longrightarrow^q_S [H']\text{let}^{q_0}\ \text{unitbe a'inb}$。（$\cdot \cdot q_0 <: 1$）并且根据规则 ST-LetUnit，$\Gamma'_{21} + \Gamma_{22} \vdash \text{let}^{q_0}\ \text{unit be a in b} :^q B$。

否则，a 是一个值。由反演可知，a = 单元类型。

于是，$[H]\text{let}^{q_0}\ \text{unit be unit in b} \Longrightarrow^q_S [H]b$。

此外，$\Gamma_{21} + \Gamma_{22} \vdash b :^q B$。（$\cdot \cdot$ 若情况为 $\mathbb{N}_=$，$\overline{\Gamma_{21}} = \overline{0}$ 且若情况为 $\mathbb{N}_\geq$，$\Gamma_{21} + \Gamma_{22} <: \Gamma_{22}$.）

• 规则 ST-LetPair。令 $A = A_1 \times A_2$，和$\Gamma, x :^{q\cdot q_0\cdot r} A_1, y :^{q\cdot q_0\cdot r} A_2 \vdash b :^q B$，于 $b :^q B$ 其中 $\Gamma_{21} \vdash a :^{q\cdot q_0}$

进一步，$H \models \Gamma_1 + (\Gamma_{21} + \Gamma_{22})$。

根据归纳假设，若 a 可以归约，则 $[H]a \Longrightarrow^q \cdot \ {}^{q_0}_{S \cup fv\ b}\ [H',]a'$，并且 $H', \models \Gamma_1 + (\Gamma'_{21} + \Gamma_{22})$，且 $\Gamma'_{21} \vdash a' :^q \cdot$

$q_0{}^r A_1 \times A_2$。

那么，$[H]$令$_{q_0}(x^r, y)$ 为 a 在 b 中 $\Longrightarrow^q \cdot \ {}_S{}^{q_0}$ $[H',]$令$_{q_0}(x^r, y)$ 为 a′在 b 中。

根据规则 HeapStep-Discard，$[H]$令$_{q_0}(x^r, y)$ 为 a，且 b $\Longrightarrow^q_S [H',]$令$_{q_0}(x^r, y)$ 为 a′，且 b。（$\cdot \cdot q_0 <: 1$）

并且根据规则ST-LetPair，$\Gamma'_{21} + \Gamma_{22} \vdash$ 令$_{q_0}(x^r, y)$为a'，且b：B。

否则，a 是一个值。根据反演，$a = (a_1{}^r, a_2)$。

此外，$\exists \Gamma_{211}, \Gamma_{212}$ 使得 $\Gamma_{211} \vdash a_1 :^q \cdot^{q_0\cdot r} A_1$ 和 $\Gamma_{212} \vdash a_2 :^q \cdot$

$^{q_0} A_2$ 和 $\Gamma_{21} = \Gamma_{211} + \Gamma_{212}$。现在，$[H]$令$_{q_0}(x^r, y)$ 为 $(a_1{}^r, a_2)$，并在 b $\Longrightarrow^q_S [H, x \overset{q\ q_0\ r}{\mapsto} \cdot\ a_1, y \overset{q\ q_0}{\mapsto} \cdot\ a_2]b$（假设 x，y 是新鲜的）。并且，$H, x \overset{q\ q_0\ r}{\mapsto} \cdot\ a_1, y \overset{q\ q_0}{\mapsto} \cdot\ a_2 \models (\Gamma_1 + \Gamma_{22}), x :^{q\cdot q_0\cdot r} A_1, y :^{q\cdot q_0} A_2$。

• 规则 ST-Case。已知：$\Gamma_{21} + \Gamma_{22} \vdash \text{case}^{q_0}\ a\ \text{of x1.b1 ; x2.b2} :^q B$，其中 $\Gamma_{21} \vdash a :^{q\cdot q_0} A_1 + A2$，且 $\Gamma_{22}, x1 :^{q\cdot q_0} A_1 \vdash b1 :^q B$，且 $\Gamma_{22}, x2 :^{q\cdot q_0} A_2 \vdash b2 :^q B$。此外，$H \models \Gamma_1 + (\Gamma_{21} + \Gamma_{22})$。根据归纳假设，如果 a 发生步进，则 $[H]a \Longrightarrow^{q\cdot q_0}_{S \cup fv\ b_1 \cup fv\ b_2} [H',]a'$，并且 $H' \models \Gamma_1 + (\Gamma'_{21} + \Gamma_{22})$，且 $\Gamma'_{21} \vdash a' :^{q\cdot q_0\ r} A_1 \times A2$。于是，$[H]\text{case}^{q_0}\ a\ \text{of x1.b1 ; x2.b}_2 \Longrightarrow^{q\cdot}_{S\ q_0} [H']\text{case}^{q_0}\ a'\ \text{of x1.b1 ; x2.b2}$。根据规则 HeapStep-Discard，$[H]\ \text{case}^{q_0}\ a\ \text{of x1.b1 ; x2.b}_2 \Longrightarrow^q_S [H']\text{case}^{q_0}\ a'\ \text{of x1.b1 ; x2.b2}$。（$\cdot \cdot q_0 <: 1$）并且根据规则 ST-Case，$\Gamma'_{21} + \Gamma_{22} \vdash \text{case}^{q_0}\ a'\ \text{of x1.b1 ; x2.b2} :^q B$。否则，a 是一个值。通过反演，$a = \text{inj1 a1}$ 或 $a = \text{inj2 a2}$。设 $a = \text{inj1 a1}$。于是，$\Gamma_{21} \vdash a1 :^{q\cdot q_0} A1$。现在，$[H]\ \text{case}^{q_0}(\text{inj1 a1})\ \text{of x1.b1 ; x2.b}_2 \Longrightarrow^q_S [H, x_1 \overset{q\ q_0}{\mapsto} a_1]b1$（假设 x1 是新鲜的）。

并且，H，$x_1 \overset{q}{\mapsto}{}^{q_0} a_1 \models (\Gamma_1 + \Gamma_{22})$，$x_1 :$ $^{q\cdot q_0} A_1$。$a = \mathrm{inj}_2\, a_2$ 的情况类似。

- 规则 ST-SubL。已知：$\Gamma_2 \vdash a :^q A$，其中 $\Gamma'_2 \vdash a :^q A$，且 $\Gamma_2 <: \Gamma'2$。此外，$H \models \Gamma_1 + \Gamma_2$。由于 $\Gamma_2 <: \Gamma'2$，存在 Γ_0 使得 $\Gamma_2 = \Gamma'_2 + \Gamma_0$。（对于 $\mathbb{N}_=$，$\Gamma_0 = \overline{0}$ 的情况。）根据归纳假设，如果 a 不是值，那么 $[H]a \Longrightarrow^q_S [H']a'$，且 $H' \models (\Gamma_1 + \Gamma_0) + \Gamma''2$，并且 $\Gamma''_2 \vdash a' :^q A$。于是，根据规则 ST-SubL，$\Gamma_0 + \Gamma''_2 \vdash a' :^q A$，因为 $\Gamma_0 + \Gamma''_2 <: \Gamma''2$。
- 规则 ST-SubR。已知 $\Gamma_2 \vdash a :^{q'} A$，其中 $\Gamma_2 \vdash a :^q A$，且 $q <: q'$。此外，$H \models \Gamma_1 + \Gamma_2$。根据归纳假设，如果 a 不是值，那么 $[H]a \Longrightarrow^q_S [H']a'$，且 $H' \models \Gamma_1 + \Gamma'2$，并且 $\Gamma'_2 \vdash a' :^q A$。然后，由于 $q <: q'$，根据规则 HeapStep-Discard，$[H]a \Longrightarrow^{q'}{}_S [H']a'$。并且根据规则 ST-SubR，$\Gamma'_2 \vdash a' :^{q'} A$。

□

定理 D.5（可靠性（定理 4.3））。如果 $H \models \Gamma$ 且 $\Gamma \vdash a :^q A$ 且 $q \neq 0$，那么 a 要么是一个值，要么存在 H'，Γ', a'，使得 $[H]a \Longrightarrow^q_S [H',]a'$ 且 $H' \models \Gamma'$ 且 $\Gamma' \vdash a' :^q A$。

证明。使用引理 D.4，其中 $\Gamma_1 := 0 \cdot \Gamma$ 以及 $\Gamma_2 := \Gamma$。 □

引理 D.6。如果 $H \models \Gamma_1 \sqcap \Gamma_2$ 且 $\Gamma_2 \vdash a :^\ell A$ 且 $l \neq \top$，那么 a 要么是一个值，要么存在 H'，Γ'_2, a'，使得：

- $[H]\alpha \Longrightarrow^\ell_S [H',]\alpha'$
- $H' \models \Gamma_1 \sqcap \Gamma'2$
- $\Gamma'_2 \vdash a' :^\ell A$

证明。对 $\Gamma_2 \vdash a :^\ell A$ 进行归纳。应用引理 D.4。 □

定理 D.7（可靠性（定理 4.3））。若 $H \models \Gamma$ 且 $\Gamma \vdash a :^\ell A$ 且 $l \neq \top$，则 a 要么是一个值，要么存在 H'，Γ'、a'，使得 $[H]a \Longrightarrow^\ell_S [H',]a'$，且 $H' \models \Gamma'$ 且 $\Gamma' \vdash a' :^\ell A$。

证明。应用引理 D.6，取 $\Gamma_1 := \top \sqcup \Gamma$ 和 $\Gamma_2 := \Gamma$。 □

推论 D.8（无用法（推论 4.4））。在 *LDC($\mathbb{Q}_\mathbb{N}$)* 中：设 $\emptyset \vdash f :^{1\ 0} A \to A$。那么，对于任意 $\emptyset \vdash a_1 :^0 A$ 和 $\emptyset \vdash a_2 :^0 A$，项 $f\, a_1{}^0$ 和 $f\, a_2{}^0$ 具有相同的操作行为。

证明。为了说明原因，我们考察 $f\, a_1{}^0$ 和 $f\, a_2{}^0$ 的归约。

令 $[H]a \Longrightarrow^q_j [H',]a'$ 表示进行 j 步的归约，其中 H 和 a 分别是初始堆和项，H′ 和 a′ 分别是最终堆和项，q 是发生归约的位置标签。对于某些 j、H 和 b，我们有：$[\emptyset]f\, a_1{}^0 \Longrightarrow^1_j [H](\lambda^0 x.b)\, a_1{}^0$ 和 $[\emptyset]f\, a_2{}^0 \Longrightarrow^1_j [H](\lambda^0 x.b)\, a_2{}^0$。

于是，$[H](\lambda^0 x.b) a_1{}^0 \Longrightarrow^1_S [H, x \overset{0}{\mapsto} a_1]b$，且 $[H](\lambda^0 x.b)\, a_2{}^0 \Longrightarrow^1_S [H, x \overset{0}{\mapsto} a_2]b$（其中 x 新鲜）。

但是，如果 $[H, x \overset{0}{\mapsto} a_1]b \Longrightarrow^1_k [H', x \overset{0}{\mapsto} a_1, H'']b'$，那么对于任意 k，$[H, x \overset{0}{\mapsto} a_2]b \Longrightarrow^1_k [H', x \overset{0}{\mapsto} a_2, H'']b'$（根据引理 4.2）。

因此，$f\, a_1{}^0$ 和 $f\, a_2{}^0$ 具有相同的操作行为。

□

推论 D.9（非干扰（推论 4.5））。在 *LDC(ℒ)* 中：设 $\emptyset \vdash f :^{\mathrm{L}\ \mathrm{H}} A \to A$。那么，对于任意 $\emptyset \vdash a_1 :^{\mathrm{H}} A$ 和 $\emptyset \vdash a_2 :^{\mathrm{H}} A$，项 $f\, a_1{}^{\mathrm{H}}$ 和 $f\, a_2{}^{\mathrm{H}}$ 具有相同的操作行为。

证明。与推论 D.8 的证明相同，只需将 0 和 1 分别替换为 H 和 L。 □

推论 D.10（仿射使用（推论 4.6））。在 *LDC($\mathbb{N}_\geq$)* 中：设 $\emptyset \vdash f :^{1\ 1} A \to A$。那么，对于任意 $\emptyset \vdash a :^1 A$，项 $f\, a^1$ 在归约过程中使用 a 至多一次。

证明。要理解原因，请考察 f a[1] 的归约。

对于某些 j、H 和 b，我们有，$[\emptyset]\mathrm{f}\ \mathrm{a}^1 \Longrightarrow_j^1 [\mathrm{H}\ ](\lambda^1 \mathrm{x}\ .\mathrm{b})\ \mathrm{a}^1$。

然后，$[\mathrm{H}\ ](\lambda^1 \mathrm{x}\ 。\mathrm{b})\ \mathrm{a}^1 \Longrightarrow_S^1 [\mathrm{H}\ ,\ \mathrm{x}\ \overset{1}{\mapsto} \mathrm{a}]\mathrm{b}$（对于 x 新鲜）。

现在，b 可能在完全不查找 x 的情况下归约为一个值，也可能恰好查找一次 x 的值。使用一次 x 会将允许的使用次数从 1 降为 0，此后它基本上无法再使用。因此，a 的使用次数不能超过一次。□

E LDC 的 PTS 版本中的线性性分析

引理 E.1（乘法）。如果 $\Gamma \vdash \mathrm{a} :^q \mathrm{A}$，那么 $\mathrm{r}_0 \cdot \Gamma \vdash \mathrm{a} :^{r_0 \cdot q} \mathrm{A}$。

证明。 对 $\Gamma \vdash a$ 进行归纳：$^q A$。遵循引理B.1的证明。 □

引理 E.2（因式分解）。 如果 $\Gamma \vdash a :^q A$ 且 $q \neq 0$，那么存在 Γ' 使得 $\Gamma' \vdash a :^1 A$且 $\Gamma <: q \cdot \Gamma'$.

证明。 对 $\Gamma \vdash a$ 进行归纳：$^q A$。遵循引理B.2的证明。 □

引理 E.3（拆分）。 如果 $\Gamma \vdash a :^{q_1+q_2} A$，则存在 Γ_1 和 Γ_2 使得 $\Gamma_1 \vdash a :^{q_1} A$且$\Gamma_2 \vdash a :^{q_2} A$ 以及 $\Gamma = \Gamma_1 + \Gamma_2$。

证明。如果 $\mathrm{q}_1 + \mathrm{q}_2 = 0$，那么 $\Gamma_1 := 0 \cdot \Gamma$ and $\Gamma_2 := \Gamma$。

否则，根据引理 E.2， $\exists \Gamma'$ 使得 $\Gamma' \vdash$ 一个 : $^1 A$ 且 $\Gamma <: (q_1 + q_2) \cdot \Gamma'$。

然后，$\Gamma = (\mathrm{q}_1 + \mathrm{q}_2) \cdot \Gamma' + \Gamma_0$ 对于某个 Γ_0（在 case 表达式为 $\mathbb{N}_=$ 时，$\overline{\Gamma_0 = 0}$）。

现在，根据引理E.1，$q_1 \cdot \Gamma' \vdash a :\ ^{q_1} A$ 和 $q_2 \cdot \Gamma' \vdash a :\ ^{q_2} A$。根据规则 PTS-SubL，$q_2 \cdot \Gamma' + \Gamma_0 \vdash a :\ ^{q_2} A$。

通过令 $\Gamma_1 := \mathrm{q}_1 \cdot \Gamma'$ 和 $\Gamma_2 := \mathrm{q}_2 \cdot \Gamma' + \Gamma_0$，即可得出该引理。 □

引理 E.4（弱化）。若 $\Gamma_1,\ \Gamma_2 \vdash \mathrm{a} :^q \mathrm{A}$ 且 $\Delta_1 \vdash_0 \mathrm{C} : \mathrm{s}$ 并且 $\lfloor \Gamma_1 \rfloor = \Delta_1$，则 $\Gamma_1,\ \mathrm{z} :^0 \mathrm{C}\ \Gamma_2 \vdash \mathrm{a} :^q \mathrm{A}$。

证明。对 $\Gamma_1, \Gamma_2 \vdash \mathrm{a} :^q \mathrm{A}$ 进行归纳。 □

引理 E.5（替换（引理 5.1））。若 $\Gamma_1,\ \mathrm{z} :^{r_0} \mathrm{C}\ \Gamma_2 \vdash \mathrm{a} :^q \mathrm{A}$ 且 $\Gamma \vdash \mathrm{c} :^{r_0} \mathrm{C}$ and$\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$，则 $\Gamma_1 + \Gamma,\ \Gamma_2\{\mathrm{c/z}\} \vdash \mathrm{a}\{\mathrm{c/z}\} :^q \mathrm{A}\{\mathrm{c/z}\}$。

证明。通过对 $\Gamma_1,\ \mathrm{z} :^{r_0} \mathrm{C}\ \Gamma_2 \vdash \mathrm{a} :^q \mathrm{A}$ 进行归纳。

- 规则 PTS-Var。已知：$0 \cdot \Gamma_1,\ \mathrm{x} :^q \mathrm{A} \vdash \mathrm{x} :^q \mathrm{A}$，其中 $\Delta \vdash_0 \mathrm{A} : \mathrm{s}$ 以及 $\lfloor \Gamma_1 \rfloor = \Delta$。需要考虑两种情况。– 已知：$0 \cdot \Gamma_1,\ \mathrm{x} :^q \mathrm{A} \vdash \mathrm{x} :^q \mathrm{A}$ 以及 $\Gamma \vdash \mathrm{a} :^q \mathrm{A}$，其中 $\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$。需要证明：$\Gamma \vdash \mathrm{a} :^q \mathrm{A}$。由已知条件可得。– 已知：$0 \cdot \Gamma_{11},\ \mathrm{z} :^0 \mathrm{C},\ 0 \cdot \Gamma_{12},\ \mathrm{x} :^q \mathrm{A} \vdash \mathrm{x} :^q \mathrm{A}$ 以及 $\Gamma \vdash \mathrm{c} :^0 \mathrm{C}$，其中 $\lfloor \Gamma_{11} \rfloor = \lfloor \Gamma \rfloor$。需要证明：$\Gamma,\ 0 \cdot \Gamma_{12}\{\mathrm{c/z}\},\ \mathrm{x} :^q \mathrm{A}\{\mathrm{c/z}\} \vdash \mathrm{x} :^q \mathrm{A}\{\mathrm{c/z}\}$。根据规则 PTS-Var 和 PTS-SubL 可得（注意，在 $\mathbb{N}_=, \Gamma = \overline{0}$ 的情况下）。

- PTS-Weak规则。已知：$\Gamma_1, \mathrm{y} :^0 \mathrm{B} \vdash \mathrm{a} :^q \mathrm{A}$，其中$\Gamma_1 \vdash \mathrm{a} :^q \mathrm{A}$且$\Delta \vdash_0 \mathrm{B} : \mathrm{s}$和 $\lfloor \Gamma_1 \rfloor = \Delta$。需要考虑两种情况。– 已知：$\Gamma_1, \mathrm{y} :^0 \mathrm{B} \vdash \mathrm{a} :^q \mathrm{A}$以及 $\Gamma \vdash \mathrm{b} :^0 \mathrm{B}$，其中 $\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$。需证明：$\Gamma_1 + \Gamma \vdash \mathrm{a}\{\mathrm{b/y}\} :^q \mathrm{A}\{\mathrm{b/y}\}$。由于 $y \notin \mathrm{fv}\ a$和 $y \notin \mathrm{fv}\ A$，需证明：$\Gamma_1 + \Gamma \vdash \mathrm{a} :^q \mathrm{A}$。根据 PTS-SubL规则可得此情况（注意，在 $\mathbb{N}_=$、$\Gamma = \overline{0}$ 的case 表达式中）。– 已知：$\Gamma_{11}, \mathrm{z} :^{r_0} \mathrm{C}\ \Gamma_{12}, \mathrm{y} :^0 \mathrm{B} \vdash \mathrm{a} :^q \mathrm{A}$以及 $\Gamma \vdash \mathrm{c} :^{r_0} \mathrm{C}$，其中 $\lfloor \Gamma_{11} \rfloor = \lfloor \Gamma \rfloor$。需证明：$\Gamma_{11} + \Gamma, \Gamma_{12}\{\mathrm{c/z}\}, \mathrm{y} :^0 \mathrm{B}\ \{\mathrm{c/z}\} \vdash \mathrm{a}\{\mathrm{c/z}\} :^q \mathrm{A}\{\mathrm{c/z}\}$。由归纳假设和 PTS-Weak规则可得。

• 规则 PTS-Pi。已知： $\Gamma_{11}+\Gamma_{21}, z:^{r_{01}+r_{02}} C\ \ \Gamma_{12}+\Gamma_{22} \vdash \Pi x:^{r} A.B :^{q} s3$，其中 $\Gamma_{11}, z:^{r_{01}} C\ \ \Gamma_{12} \vdash A :^{q} s1$ 且 $\Gamma_{21}, z:^{r_{02}} C\ \ \Gamma_{22}, x:^{q_0} A \vdash B :^{q} s2$，并且 R(s1, s2, s3)。此外， $\Gamma \vdash c :^{r_{01}+r_{02}} C$，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_{11}\rfloor$。需要证明： $\Gamma_{11}+\Gamma_{21}+\Gamma, \Gamma_{12}\{c/z\} + \Gamma_{22}\{c/z\} \vdash \Pi x:^{r} A\{c/z\}.B\{c/z\} :^{q} s3$。由引理 E.3， $\exists\Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c :^{r_{01}} C$ 且 $\Gamma_{32} \vdash c :^{r_{02}} C$，并且 $\Gamma_{31}+\Gamma_{32} = \Gamma$。由归纳假设， $\Gamma_{11}+\Gamma_{31}, \Gamma_{12}\{c/z\} \vdash A\{c/z\} :^{q} s1$ 且 $\Gamma_{21}+\Gamma_{32}, \Gamma_{22}\{c/z\}, x:^{q_0} A\{c/z\} \vdash B\{c/z\} :^{q} s2$。因此，此 case 表达式由规则 PTS-Pi 得证。

• 规则 PTS-Lam。已知： $\Gamma_1,\ z:^{r_0} C\ \ \Gamma_2 \vdash \lambda^{r} x :A.b:^{q} \Pi x:^{r} A.B$，其中 $\Gamma_1,\ z:^{r_0} C\ \ \Gamma_2, x:^{q\cdot r} A \vdash b :^{q} B$。此外， $\Gamma \vdash c :^{r_0} C$，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_1\rfloor$。需要证明： $\Gamma_1+\Gamma, \Gamma_2\{c/z\} \vdash \lambda^{r} x:A\{c/z\}.b\{c/z\} :^{q} \Pi x:^{r} A\{c/z\}.B\{c/z\}$。根据归纳假设， $\Gamma_1+\Gamma, \Gamma_2\{c/z\}, x:^{q\cdot r} A\{c/z\} \vdash b\{c/z\} :^{q} B\{c/z\}$。因此，此 case 表达式由规则 PTS-Lam 得出。

• PTS-App规则。已知： $\Gamma_{11}+\Gamma_{21}, z:^{r_{01}+r_{02}} C\ \ \Gamma_{12}+\Gamma_{22} \vdash ba^{r} :^{q} B\{a/x\}$，其中 $\Gamma_{11}, z:^{r_{01}} C\ \ \Gamma_{12} \vdash b:^{q} \Pi x :^{r} A.B$ 且 $\Gamma_{21}, z:^{r_{02}} C\ \ \Gamma_{22} \vdash a :^{q\cdot r} A$。此外， $\Gamma \vdash c :^{r_{01}+r_{02}} C$，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_{11}\rfloor$。需证明： $\Gamma_{11}+\Gamma_{21}+\Gamma, \Gamma_{12}\{c/z\} + \Gamma_{22}\{c/z\} \vdash b\{c/z\}\ a\{c/z\}^{r} :^{q} B\{c/z\}\{a\{c/z\}/x\}$。根据引理E.3， $\exists\Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c :^{r_{01}} C$且 $\Gamma_{32} \vdash c:^{r_{02}} C$，并且 $\Gamma_{31}+\Gamma_{32} = \Gamma$。根据归纳假设， $\Gamma_{11}+\Gamma_{31}, \Gamma_{12}\{c/z\} \vdash b\{c/z\} :^{q} \Pi x :^{r} A\{c/z\}.B\{c/z\}$ 且 $\Gamma_{21}+\Gamma_{32}, \Gamma_{22}\{c/z\} \vdash a\{c/z\}:^{q\cdot r} A\{c/z\}$。因此，此 case 表达式由 PTS-App规则得证。

PTS-Conv规则。给定： •, $z :^{r_0} C\ \ \Gamma_2 \vdash a :^{q} B$，其中 $\Gamma_1, z :^{r_0} C\ \ \Gamma_2 \vdash a:^{q} A$ 且 $A =_{\beta} B$。进一步地， $\Gamma \vdash c :^{r_0} C$，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_1\rfloor$。需要证明： $\Gamma_{11}+\Gamma, \Gamma_2\{c/z\} \vdash a\{c/z\} :^{q} B\{c/z\}$。根据归纳假设， $\Gamma_{11}+\Gamma, \Gamma_2\{c/z\} \vdash a\{c/z\} :^{q} A\{c/z\}$。此外，由于 $A =_{\beta} B$，所以 $A\{c/z\} =_{\beta} B\{c/z\}$。因此，这个 case 表达式即可由 PTS-Conv 规则推出。

• PTS-Pair规则。给定： $\Gamma_{11}+\Gamma_{21}, z:^{r_{01}+r_{02}} C\ \ \Gamma_{12}+\Gamma_{22} \vdash (a_1^{r}, a2):^{q} \Sigma x :^{r} A1.A2$，其中 $\Gamma_{11}, z :^{r_{01}} C\ \ \Gamma_{12} \vdash a1:^{q\cdot r} A1$ 且 $\Gamma_{21}, z:^{r_{02}} C\ \ \Gamma_{22} \vdash a2:^{q} A2\{a1/x\}$。此外， $\Gamma \vdash c :^{r_{01}+r_{02}} C$，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_{11}\rfloor$。需要证明： $\Gamma_{11}+\Gamma_{21}+\Gamma, \Gamma_{12}\{c/z\}+\Gamma_{22}\{c/z\} \vdash (a1\{c/z\}^{r}, a2\{c/z\}):^{q} \Sigma x:^{r} A1\{c/z\}.A2\{c/z\}$。根据引理E.3， $\exists\Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c :^{r_{01}} C$且 $\Gamma_{32} \vdash c:^{r_{02}} C$，并且 $\Gamma = \Gamma_{31}+\Gamma_{32}$。根据归纳假设， $\Gamma_{11}+\Gamma_{31}, \Gamma_{12}\{c/z\} \vdash a1\{c/z\} :^{q\cdot r} A1\{c/z\}$ 且 $\Gamma_{21}+\Gamma_{32}, \Gamma_{22}\{c/z\} \vdash a2\{c/z\} :^{q} A2\{a1/x\}\{c/z\}$。现在， $A2\{a1/x\}\{c/z\} = A2\{c/z\}\{a1\{c/z\}/x\}$。因此，这个 case 表达式可由 PTS-Pair规则推出。

• PTS-LetPair 规则。给定： $\Gamma_{11}+\Gamma_{21}, z :^{r_{01}+r_{02}} C\ \ \Gamma_{12}+\Gamma_{22} \vdash let_{q_0}(x^{r},y)\ bea\ in\ b:^{q} B\{a/w\}$，其中 $\Delta, w: \Sigma x :^{r} A1.A_2 \vdash_0 B$: s且 $\Gamma_{11}, z:^{r_{01}} C\ \ \Gamma_{12} \vdash a :^{q\cdot q_0} \Sigma x :^{r} A1.A2$，并且 $\Delta = \lfloor\Gamma_{11}\rfloor, z : C\ \lfloor\Gamma_{12}\rfloor$ 以及 $\Gamma_{21}, z:^{r_{02}} C\ \ \Gamma_{22}, x:^{q\cdot_{q_0}\cdot r} A1,\ y :^{q\cdot q_0} A_2 \vdash b:^{q} B\{(x^{r}, y)/w\}$。此外， $\Gamma \vdash c :^{r_{01}+r_{02}} C$，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_{11}\rfloor$。需要证明： $\Gamma_{11}+\Gamma_{21}+\Gamma, \Gamma_{12}\{c/z\} + \Gamma_{22}\{c/z\} \vdash let_{q_0}(x^{r},y)be\ a\{c/z\}\ in\ b\{c/z\} :^{q} B\{c/z\}\{a\{c/z\}/w\}$。根据引理E.3， $\exists\Gamma_{31}, \Gamma_{32}$ 使得 $\Gamma_{31} \vdash c :^{r_{01}} C$且 $\Gamma_{32} \vdash c:^{r_{02}} C$，并且 $\Gamma = \Gamma_{31}+\Gamma_{32}$。根据归纳假设， $\Gamma_{11}+\Gamma_{31}, \Gamma_{12}\{c/z\} \vdash a\{c/z\}:^{q\cdot q_0} \Sigma x :^{r} A1\{c/z\}.A2\{c/z\}$且 $\Gamma_{21}+\Gamma_{32}, \Gamma_{22}\{c/z\}, x:^{q\cdot_{q_0}\cdot r} A1\{c/z\}, y :^{q\cdot q_0} A2\{c/z\} \vdash b\{c/z\} :^{q} B\{(x^{r}, y)/w\}\{c/z\}$。因此，此 case 表达式由 PTS-LetPair 规则得出。

- 规则 PTS-Sum、 PTS-Inj1以及 PTS-Inj2。根据归纳假设。

- PTS-Case 规则。已知： $\Gamma_{11}+\Gamma_{21}, z :^{r_{01}+r_{02}} C\ \ \Gamma_{12}+\Gamma_{22} \vdash$ case 表达式^{q_0} a of x1.b1 ; x2.b2 :^{q}B {a/w}，其中Δ, w: $A_1+A_2 \vdash_0 B : s$，且 $\Gamma_{11}, z :^{r_{01}} C\ \ \Gamma_{12} \vdash a :^{q\cdot q_0} A_1+A2$，$\Delta = \lfloor\Gamma_{11}\rfloor, z : C\ \ \lfloor\Gamma_{12}\rfloor$，以及$\Gamma_{21}, z :^{r_{02}} C\ \ \Gamma_{22}, x1 :^{q\cdot q_0} A_1 \vdash b1 :^{q} B$ {inj1 x1/w}，以及 $\Gamma_{21}, z :^{r_{02}} C\ \ \Gamma_{22}, x2 :^{q\cdot q_0} A_2 \vdash b2 :^{q} B$ {inj2 x2/w}。此外，$\Gamma \vdash c :^{r_{01}+r_{02}} C$，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_{11}\rfloor$。需证明：$\Gamma_{11}+\Gamma_{21}+\Gamma, \Gamma_{12}${c/z }+$\Gamma_{22}${c/z } $\vdash$ case 表达式^{q_0} a{c/z }of x1.b1{c/z }; x2.b2{c/z } :^{q}B {c/z }{a{c/z }/w}。由引理 E.3，$\exists\Gamma_{31}, \Gamma_{32}$使得 $\Gamma_{31} \vdash c :^{r_{01}} C$ 且 $\Gamma_{32} \vdash c :^{r_{02}} C$，并且 $\Gamma = \Gamma_{31}+\Gamma_{32}$。根据归纳假设，$\Gamma_{11}+\Gamma_{31}, \Gamma_{12}${c/z } $\vdash$ a{c/z } :$^{q\cdot q_0}$ A1{c/z } + A2{c/z }，且 $\Gamma_{21}+\Gamma_{32}, \Gamma_{22}${c/z }, x1 :$^{q\cdot q_0}$ A1{c/z } $\vdash$ b1{c/z } :^{q}B {inj1 x1/w}{c/z }，以及 $\Gamma_{21}+\Gamma_{32}, \Gamma_{22}${c/z }, x2 :$^{q\cdot q_0}$ A2{c/z } $\vdash$ b2{c/z } :^{q}B {inj2 x2/w}{c/z }。因此，该情形可由 PTS-Case 规则推出。

- 规则 PTS-SubL 和 PTS-SubR。根据归纳假设。

□

Theo注 E.6（保持性（定理 5.2））。若 $\Gamma \vdash a :\ ^{q} A$ 且 $\vdash a \rightsquigarrow a'$，则 $\Gamma \vdash a' :\ ^{q} A$.

证明。通过对 $\Gamma \vdash$ a :^{q} A 进行归纳，并对 $\vdash$ a $\rightsquigarrow$ a′ 进行反演。

- 规则 PTS-应用. 给定： $\Gamma_1+\Gamma_2 \vdash b\ a^{r} :^{q} B\{a/x\}$ 其中 $\Gamma_1 \vdash b :^{q} \Pi x :^{r} A.B$ 且 $\Gamma_2 \vdash a :^{q}$ ^{r} A.设 $\vdash$ ba$^{r} \rightsquigarrow$ c. 通过反演：$-\vdash$ b a$^{r} \rightsquigarrow$ b′ a^{r}，当 $\vdash$ b $\rightsquigarrow$ b′。需要证明：$\Gamma_1+\Gamma_2 \vdash$ b′ a^{r}:^{q}B {a/x }。由归纳假设和PTS-App规则可得。–b $= \lambda^{r}$x: A′.b′ 并且 $\vdash$ ba$^{r} \rightsquigarrow$ b′{a/x}。需要证明：$\Gamma_1+\Gamma_2 \vdash$ b′{a/x }:^{q}B {a/x}。对 $\Gamma_1 \vdash \lambda^{r}$x: A′.b′:^{q} Πx:^{r} A.B 进行反演，可得 Γ_1，x :$^{q_0\cdot r}$ A″ $\vdash$ b′:^{q_0}B，其中 $q_0 <: q$ 且 $A'' =_{\beta} A$。现在，需要考虑两个 case 表达式。

  - $q_0 = 0$。由于 $q_0 <: q$，因此 $q = 0$。接着，根据 PTS-Conv 规则和替换引理，$\Gamma_1+\Gamma_2 \vdash$ b, {a/x } :^{0} B {a/x }。
  - $q_0 \neq 0$。根据引理 E.2，存在 $\exists\Gamma'1$ 和 r′，使得 $\Gamma'1, x :^{r'} A''\ \ \vdash b' :^{1} B$，且 $\Gamma_1 <: q0\cdot\Gamma'1$，并且 $q0\cdot r <: q0\cdot r'$。由于 $q_0 \neq 0$，因此 $r <: r'$。于是，根据 PTS-SubL规则，$\Gamma'1, x :^{r} A''\ \ \vdash b' :^{1} B$。现在，根据引理 E.1，$q\cdot\Gamma'1, x :^{q\cdot r} A''\ \ \vdash b' :^{q} B$。根据 PTS-SubL规则，$\Gamma_1, x :^{q\cdot r} A''\ \ \vdash b' :^{q} B$。于是，此 case 表达式可由 PTS-Conv 规则和替换引理推出。

- 规则 PTS-LetUnit。已知：$\Gamma_1+\Gamma_2 \vdash$ let^{q_0} unit be a in b :^{q}B {a/z }，其中 Δ，z : 单元类型 $\vdash_0$ B : s，以及 $\Delta = \lfloor\Gamma_1\rfloor$和 $\Gamma_1 \vdash a :^{q\cdot q_0}$ 单元类型，且 $\Gamma_2 \vdash b :^{q} B$ {unit/z }。令 $\vdash$ let ^{q_0} unit be a in b $\rightsquigarrow$ c。通过反演：$-\vdash$ let^{q_0} unit be a in b $\rightsquigarrow$ let^{q_0} unit be a′ in b，当 $\vdash$ a $\rightsquigarrow$ a′ 时。需要证明：$\Gamma_1+\Gamma_2 \vdash$ let^{q_0} unit be a′ in b :^{q}B {a/z }。由归纳假设以及规则 PTS-LetUnit 和 PTS-Conv 得出。$-\vdash$ let^{q_0} unit be unit in b $\rightsquigarrow$ b。需要证明：$\Gamma_1+\Gamma_2 \vdash$ b :^{q}B {unit/z }。由规则 PTS-SubL 得出（注意在 $\mathbb{N}_{=}$, $\Gamma_1 = 0$ 的情况下）。

• PTS-LetPair 规则。已知： $\Gamma_1 + \Gamma_2 \vdash \text{let}_{q_0}(x\ ^r, y)$ be a in b $:^q B\ \{a/z\}$，其中 Δ，z : $\Sigma x :^r A1.A_2 \vdash_0 B : s$，且 $\Delta = \lfloor \Gamma_1 \rfloor$ 以及 $\Gamma_1 \vdash a :^{q \cdot q_0} \Sigma x :^r A1.A2$，且 Γ_2，x $:^{q \cdot_{q_0} r} A1$，y $:^{q \cdot q_0} A_2 \vdash b :^q B\ \{(x\ ^r, y)/z\}$。令 $\vdash \text{let}_{q_0}(x^r, y)$ be a in b $\rightsquigarrow c$。根据反演： $- \vdash \text{let}_{q_0}(x\ ^r, y)$ be a in b $\rightsquigarrow \text{let}_{q_0}(x^r, y)$ be a′ in b，当 $\vdash a \rightsquigarrow a'$ 时。需要证明： $\Gamma_1 + \Gamma_2 \vdash \text{let}_{q_0}(x^r, y)$ be a′ in b $:^q B\ \{a/z\}$。由归纳假设以及 PTS-LetPair 和 PTS-Conv 规则可得。$- \vdash \text{let}_{q_0}(x\ ^r, y)$ be $(a_1{}^r, a2)$ in b $\rightsquigarrow b\{a1/x\}\{a2/y\}$。需要证明： $\Gamma_1 + \Gamma_2 \vdash b\{a1/x\}\{a2/y\} :^q B\ \{(a_1{}^r, a2)/z\}$。对 $\Gamma_1 \vdash (a_1{}^r, a2) :^{q \cdot q_0} \Sigma x :^r A1.A2$ 进行反演，我们有：$\exists \Gamma_{11}$，Γ_{12} 使得 $\Gamma_{11} \vdash a1 :^{q \cdot_{q_0} r} A1$，且 $\Gamma_{12} \vdash a2 :^{q \cdot q_0} A2\{a1/x\}$，以及 $\Gamma_1 = \Gamma_{11} + \Gamma_{12}$。应用替换引理，我们得到，$\Gamma_2 + \Gamma_{11}$，y $:^{q \cdot q_0} A2\{a1/x\} \vdash b\{a1/x\} :^q B\ \{(a_1{}^r, y)/z\}$。再次应用该引理，我们得到，$\Gamma_2 + \Gamma_{11} + \Gamma_{12} \vdash b\{a1/x\}\{a2/y\} :^q B\ \{(a_1{}^r, a2)/z\}$，证毕。

• PTS-Case 规则。已知： $\Gamma_1 + \Gamma_2 \vdash \text{case}_{q_0}$ a of x1.b1 ; x2.b2 $:^q B\ \{a/z\}$，其中 Δ，z : $A_1 + A_2 \vdash_0 B : s$，且 $\Delta = \lfloor \Gamma_1 \rfloor$ 以及 $\Gamma_1 \vdash a :^{q \cdot q_0} A_1 + A2$，且 Γ_2，x1 $:^{q \cdot q_0} A_1 \vdash b1 :^q B\ \{\text{inj1 x1}/z\}$，且 Γ_2，x2 $:^{q \cdot q_0} A_2 \vdash b2 :^q B\ \{\text{inj2 x2}/z\}$。令 $\vdash \text{case}_{q_0}$ a of x1.b1 ; x2.$b_2 \rightsquigarrow c$。通过反演： $- \vdash \text{case}_{q_0}$ a of x1.b1 ; x2.$b_2 \rightsquigarrow \text{case}_{q_0}$ a′ of x1.b1 ; x2.b2，其中 $\vdash a \rightsquigarrow a'$。需要证明： $\Gamma_1 + \Gamma_2 \vdash \text{case}_{q_0}$ a′ of x1.b1 ; x2.b2 $:^q B\ \{a/z\}$。由归纳假设以及 PTS-Case 和 PTS-Conv 规则可得。$- \vdash \text{case}_{q_0}$ (inj1 a1) of x1.b1 ; x2.$b_2 \rightsquigarrow$ b1{a1/x1}。需要证明 $\Gamma_1 + \Gamma_2 \vdash b1\{a1/x1\} :^q B\ \{\text{inj1 a1}/z\}$。对 $\Gamma_1 \vdash \text{inj1 a1} :^{q \cdot q_0} A_1 + A2$ 进行反演，可得 $\Gamma_1 \vdash a1 :^{q \cdot q_0} A1$。因此，这种情况可通过应用替换引理得出。$- \vdash \text{case}_{q_0}$ (inj2 a2) of x1.b1 ; x2.$b_2 \rightsquigarrow$ b2{a2/x2}。与前一种情况类似。

• 规则 PTS-Weak、 PTS-Conv 规则、PTS-SubL和 PTS-SubR。由归纳假设得出。

□

定理 E.7（进展（定理5.3））。若 $\emptyset \vdash a :^q A$，则 a 要么是一个值，要么存在 a'使得 $\vdash a \rightsquigarrow a'$。

证明。对 $\emptyset \vdash a :^q A$ 进行归纳。

• 规则 PTS-应用。已知： $\emptyset \vdash b\ a^r :^q B\ \{a/x\}$，其中 $\emptyset \vdash b :^q \Pi x :^r A$。B 且 $\emptyset \vdash a :^{q \cdot r} A$。需证明： $\exists c$，$\vdash b\ a^r \rightsquigarrow c$。根据归纳假设，b 要么是一个值，要么 $\vdash b \rightsquigarrow b'$。若 b 是一个值，则由反演可知，存在某个 b′，使得 b $= \lambda^r x : A'$。b′。因此，$\vdash b\ a^r \rightsquigarrow b'\ \{a/x\}$。否则，$\vdash b\ a^r \rightsquigarrow b'\ a^r$。

• 规则 PTS-LetUnit。已知： $\emptyset \vdash \text{let}_{q_0}$ unit be a in b $:^q B\ \{a/z\}$，其中 z : Unit $\vdash_0 B : s$ 且 $\emptyset \vdash a :^{q \cdot q_0}$ Unit，$\emptyset \vdash b :^q B\ \{\text{unit}/z\}$。需证明： $\exists c, \vdash \text{let}_{q_0}$ unit be a in b $\rightsquigarrow c$。根据归纳假设，a 要么是一个值，要么是 $\vdash a \rightsquigarrow a'$。若 a 是一个值，则根据反演，a = unit。因此，$\vdash$ let $_{q_0}$ unit be a in b $\rightsquigarrow$ b。否则，$\vdash \text{let}_{q_0}$ unit be a in b $\rightsquigarrow \text{let}_{q_0}$ unit be a′ in b。

• PTS-LetPair 规则。已知： $\emptyset \vdash \text{let}_{q_0}(x\ ^r, y)$beainb $:^q B\ \{a/z\}$，其中 z : $\Sigma x :^r A1.A_2 \vdash_0 B : s$，且 $\emptyset \vdash a :^{q \cdot q_0} \Sigma x :^r A1.A2$，以及 x $:^{q \cdot_{q_0} r} A1$，y $:^{q \cdot q_0} A_2 \vdash b :^q B\ \{(x\ ^r, y)/z\}$。需要证明： $\exists c, \vdash \text{let}_{q_0}(x^r, y)$ be a in b $\rightsquigarrow c$。

上下文，$\Gamma ::= \emptyset \mid \Gamma,\ x :^q A \mid \Gamma,\ x = a :^q A$

$\boxed{\Gamma \vdash a :^q A}$ *(额外规则)*

$$\text{PTS-DefVar}\ \frac{\Delta \vdash_0 a : A \qquad \lfloor\Gamma\rfloor = \Delta}{0 \cdot \Gamma, x = a :^q A \vdash x :^q A}$$

$$\text{PTS-DefWeak}\ \frac{\Gamma \vdash a :^q A \qquad \Delta \vdash_0 b : B \qquad \lfloor\Gamma\rfloor = \Delta}{\Gamma, y = b :^0 B \vdash a :^q A}$$

$$\text{PTS-DefConv}\ \frac{\Gamma \vdash a :^q A \qquad \Delta \vdash_0 B : s \qquad A\{\Delta\} =_\beta B\{\Delta\} \qquad \lfloor\Gamma\rfloor = \Delta}{\Gamma \vdash a :^q B}$$

图 13. 带定义的类型规则

根据归纳假设，a 要么是一个值，要么是 $\vdash a \rightsquigarrow a'$。如果 a 是一个值，那么根据反演，$a = (a_1{}^r,\ a_2)$。因此，$\vdash \text{let}^{q_0}(x^r,\ y) \text{ be } a \text{ in } b \rightsquigarrow b\{a_1/x\}\{a_2/y\}$。否则，$\vdash \text{let}^{q_0}(x^r,\ y) \text{ be } a \text{ in } b \rightsquigarrow \text{let}^{q_0}(x^r,\ y) \text{ be } a' \text{ in } b$。

- 规则 ST-Case。已知：$\emptyset \vdash \text{case表达式}^{q_0} a \text{ of } x1.b1; x2.b2 :^q B\{a/z\}$，其中 $z : A_1 + A_2 \vdash_0 B : s$，且 $\emptyset \vdash a :^{q \cdot q_0} A_1 + A2$，以及 $x1 :^{q \cdot q_0} A_1 \vdash b1 :^q B\{\text{inj}1\, x1/z\}$ 和 $x2 :^{q \cdot q_0} A_2 \vdash b2 :^q B\{\text{inj}2\, x2/z\}$。需证明：$\exists c,\ \vdash \text{case表达式}^{q_0} a \text{ of } x1.b1 ; x2.b_2 \rightsquigarrow c$。由归纳假设，a 要么是一个值，要么 $\vdash a \rightsquigarrow a'$。如果 a 是一个值，那么 $a = \text{inj}1\ a1$ 或 $a = \text{inj}2\ a2$。于是，$\vdash \text{case}$ 表达式^{q_0} a of x1.b1 ; $x2.b_2 \rightsquigarrow b1\{a1/x1\}$ 或 $\vdash$ case表达式^{q_0} a of$x1.b1$; $x2.b_2 \rightsquigarrow b2\{a2/x2\}$。否则，$\vdash$ case 表达式^{q_0} a of$x1.b1$; $x2.b_2 \rightsquigarrow$ case 表达式^{q_0} a′ of x1.b1 ; x2.b2。

- 规则 PTS-Weak、PTS-Conv、PTS-SubL和 PTS-SubR。由归纳假设可得。
- 规则 PTS-Var。由于此处的上下文为空，因此不适用。
- 其余规则所判定的项都是值。

□

F LDC 的 PTS 版本的堆语义

回顾一下，要证明依赖类型系统的堆可靠性，我们需要允许在类型中延迟进行替换。为此，我们在上下文中加入定义，以模拟替换。这些定义仅用于推导类型等式，与线性性分析和依赖关系分析无关。定义的类型规则以及使用定义的转换规则如图 13 所示。我们将这些额外规则加入该演算的类型系统中。

有几点需要注意：

- 定义不会影响这些分析，正如规则 PTS-DefVar和 PTS-De-fWeak所示。

- 转换规则 PTS-DefConv会在类型中替换定义后检查类型是否相等。这里，$A\{\Delta\}$ 表示类型 A，其中 Δ 中的定义已按相反顺序替换。借助此规则，我们可以证明：$x = \text{Unit} :^0 s \vdash \lambda^1 y :^1 \Pi y :^1 \text{Unit}。\text{Unit}。$

定义使我们能够向类型系统传达被堆延迟的替换。为实现这种传达，我们还需要更新兼容性关系，如图 14所示。

这些扩展并未改变底层系统的本质特征。以下引理确立了底层系统与扩展系统之间的对应关系。为便于区分，我们将底层系统的类型判断和兼容性判断分别记作 $\Gamma \vdash^{\mathrm{u}} a :^q A$ 和 $H \models^{\mathrm{u}} \Gamma$，将扩展系统的相应判断分别记作 $\Gamma \vdash^{\mathrm{e}} a :^q A$ 和 $H \models^{\mathrm{e}} \Gamma$。

$\boxed{H \models \Gamma}$ *(更新后的兼容性)*

$$\frac{}{\emptyset \models \emptyset}\ \text{HeapCompat-DefEmpty} \qquad \frac{H \models \Gamma_1 + \Gamma_2 \quad \Gamma_2 \vdash a :^q A}{H, x \mapsto^q a \models \Gamma_1, x := a :^q A}\ \text{HeapCompat-DefCons}$$

图 14. 带定义的兼容关系

此外，对于 $H \models^u \Gamma$，令 Γ_H 为上下文 Γ，其中的变量按照 H 定义。下面的多重代换引理指出，给定扩展系统中的一个推导，对定义进行代换即可得到底层系统中的一个推导。展开引理指出，给定底层系统中的一个推导，向上下文中添加适当的定义即可得到扩展系统中的一个推导。

引理 F.1（多重代换）。如果$H \models^e \Gamma$ 且 $\Gamma \vdash^e a :^q A$，那么 $\emptyset \vdash^u a\{\Delta\} :^q A\{\Delta\}$ 其中$\Delta = \lfloor \Gamma \rfloor$。

证明。对 $H \models^e \Gamma$ 进行归纳。 □

引理 F.2（展开）。如果$H \models^u \Gamma$ 且 $\Gamma \vdash^u a :^q A$，那么$H \models^e \Gamma_H$ 且 $\Gamma_H \vdash^e a :^q A$。

证明。对 $H \models^u \Gamma$ 进行归纳。 □

接下来，我们证明扩展系统的可靠性定理。

引理 F.3。 如果$H \models \Gamma_1 + \Gamma_2$ 且 $\Gamma_2 \vdash a :^q A$ 以及 $q \neq 0$，那么 a 要么是一个值，要么存在 H'、Γ_2'、a'，使得：

- $[H]a \Longrightarrow_S^q [H']a'$
- $H' \models \Gamma_1 + \Gamma_2'$
- $\Gamma_2' \vdash a' :^q A$

(注意，$+$ 在此处具有重载含义：$\Gamma_1 + \Gamma_2$ 表示在必要时对上下文 Γ_1 和 Γ_2 进行填充后的加法。)

证明。通过对 $\Gamma_2 \vdash a :^q A$ 进行归纳。

- PTS-DefVar规则。给定：$0 \cdot \Gamma_{21}, x := a :^{q_2} A \vdash x :^{q_2} A$，其中 $\Delta \vdash_0 a : A$ 且 $\Delta = \lfloor \Gamma_{21} \rfloor$。此外，$H \models (\Gamma_{11}, x := a :^{q_1} A) + (0 \cdot \Gamma_{21}, x := a :^{q_2} A)$。根据反演，$\exists H1, \Gamma_0$ 使得 $H = H1, x \overset{q_1+q_2}{\mapsto} a$ 且 $\Gamma_0 \vdash a :^{q_1+q_2} A$。根据引理E.3，$\exists \Gamma_{01}, \Gamma_{02}$ 使得 $\Gamma_{01} \vdash a :^{q_1} A$ 且 $\Gamma_{02} \vdash a :^{q_2} A$，并且 $\Gamma_0 = \Gamma_{01} + \Gamma_{02}$。因此，我们有：$- [H1, x \overset{q_1+q_2}{\mapsto} a]x \Longrightarrow_S^{q_2} [H1, x \overset{q_1}{\mapsto} a]a$。$-H1, x \overset{q_1}{\mapsto} a \models (\Gamma_{11}, x := a :^{q_1} A) + (\Gamma_{02}, x := a :^0 A)$。$- \Gamma_{02}, x := a :^0 A \vdash a :^{q_2} A$。

- PTS-DefWeak 规则。已有：$\Gamma_{21}, y := b :^0 B \vdash a :^q A$，其中 $\Gamma_{21} \vdash a :^q A$ 且 $\Delta \vdash_0 b : B$ 且 $\Delta = \lfloor \Gamma_{21} \rfloor$。此外，$H \models (\Gamma_{11}, y := b :^{q_1} B) + (\Gamma_{21}, y := b :^0 B)$。通过反演，存在 $\exists H1, \Gamma_0$ 使得 $H = H1$，$y \overset{q_1}{\mapsto} b$ 且 $\Gamma_0 \vdash b :^{q_1} B$ 且 $H_1 \models \Gamma_{11} + \Gamma_{21} + \Gamma_0$。根据归纳假设，存在 $\exists H'1, H'2, \Gamma'21, \Gamma'22, a'$ 使得 $- [H_1]a \Longrightarrow_{S \cup \{y\}}^q [H'1, H'_2]a'$，其中 $|H'_1| = |H_1|$

$-H'1,H'_2 \models \Gamma_{11}+\Gamma_0+\Gamma'_{21},\Gamma'_{22}-\Gamma'_{21},\Gamma'_{22} \vdash a':^{q} A$ 然后，我们有，$-[H_1,y \overset{q_1}{\mapsto} b]a \Longrightarrow_S^{q} [H'1, y \overset{q_1}{\mapsto} b, H'_2]a'-H'1, y \overset{q_1}{\mapsto} b,H'_2 \models((\Gamma_{11}, y =b:^{q_1}B) +(\Gamma'_{21}, y =b:^{0}B)) \Gamma'_{22}-\Gamma'_{21}, y = b:^{0}B\ \Gamma'_{22} \vdash a':^{q}A$

- 规则 PTS-Var 和 PTS-Weak 规则。由于只要 $H \models \Gamma_1+\Gamma_2$ 成立，Γ_2 中的每个假设都是一个定义。

- PTS-App规则。已知：$\Gamma_{21}+\Gamma_{22} \vdash b\ a^{r}:^{q}B\{a/x\}$，其中 $\Gamma_{21} \vdash b:^{q} \Pi x:^{r}A.B$ 且 $\Gamma_{22} \vdash a:^{q\cdot r}A$。此外，$H \models \Gamma_1+(\Gamma_{21}+\Gamma_{22})$。若 b 能归约，则由归纳假设可得此情形。否则，b 是一个值。由反演，$b=\lambda^{r}x:A'.b'$。此外，存在 $\exists A1,B1$，使得 $\Gamma_{21},x:^{q\cdot r}A_1 \vdash b':^{q}B1$ 且 $\Pi x:^{r}A1.B1\{\Delta_2\} =_\beta \Pi x:^{r}A.B\{\Delta_2\}$，其中 $\Delta_2=\lfloor\Gamma_{21}\rfloor$。于是，由 PTS-DefConv规则，$\Gamma_{22} \vdash a:^{q\cdot r}A1$。此外，$\Gamma_{21},x=a:^{q\cdot r}A_1 \vdash b':^{q}B1$。现在，$-[H](\lambda^{r}x:A'.b')\ a^{r} \Longrightarrow_S^{q} [H,x \overset{q\ r}{\mapsto} \cdot\ a]b'-H,x \overset{q\ r}{\mapsto} \cdot\ a \models(\Gamma_1+\Gamma_{21}),x=a:^{q\cdot r}A1-\Gamma_{21},x=a:^{q\cdot r}A_1 \vdash b':^{q}B\{a/x\}$，由 PTS-DefConv规则。

- 规则 PTS-DefConv。已知：$\Gamma_2 \vdash a:^{q} B$，其中 $\Gamma_2 \vdash a:^{q} A$ 且 $A\{\Delta_2\} =_\beta B\{\Delta_2\}$ 并且 $\Delta_2=\lfloor\Gamma_2\rfloor$。此外，$H \models \Gamma_1+\Gamma_2$。根据归纳假设，$[H]a \Longrightarrow_S^{q} [H']a'$ 且 $H' \models \Gamma_1+\Gamma'2$，并且 $\Gamma'_2 \vdash a':^{q}A$。令 $\Delta'_2=\lfloor\Gamma'_2\rfloor$。于是，$A\{\Delta'2\}=A\{\Delta_2\} =_\beta B\{\Delta_2\}=B\{\Delta'2\}$ ($\because fvA, fvB \in dom\Delta_2 \subseteq dom\Delta'2$)。因此，本情形可由规则 PTS-DefConv 得出。

- PTS-LetPair 规则。已知：$\Gamma_{21}+\Gamma_{22} \vdash let_{q_0}(x^{r},y)\ be\ a\ in\ b:^{q}B\{a/z\}$，其中 Δ_2，$z:\Sigma x:^{r}A1.A_2 \vdash_0 B:s$，且 $\Gamma_{21} \vdash a:^{q\cdot q_0}\Sigma x:^{r}A1.A2$，且 Γ_{22}，$x:^{q\cdot_{q_0}r}A1$，$y:^{q\cdot q_0}A_2 \vdash b:^{q}B\{(x^{r},y)/z\}$，且 $\Delta_2=\lfloor\Gamma_{21}\rfloor$。此外，$H \models \Gamma_1+(\Gamma_{21}+\Gamma_{22})$。若 a 进行一步归约，则 $[H]a \Longrightarrow_{S\cup fv\ b}^{q\cdot q_0} \lceil H'\rceil a'$，且 $H' \models \Gamma_1+(\Gamma'_{21}+\Gamma_{22})$，并且 $\Gamma'_{21} \vdash a':^{q\cdot q_0}\Sigma x:^{r}A1.A2$。于是，$-[H]let_{q_0}(x^{r},y)\ be\ a\ in\ b \Longrightarrow_S^{q\cdot_{q_0}} [H']let_{q_0}(x^{r},y)\ be\ a'\ in\ b$。根据规则 HeapStep-Discard，$[H]let_{q_0}(x^{r},y)\ be\ a\ in\ b \Longrightarrow_S^{q} [H']let_{q_0}(x^{r},y)\ be\ a'\ in\ b$。（$\because q_0<:1$）$-H' \models \Gamma_1+(\Gamma'_{21}+\Gamma_{22})$ – 根据规则 ST-LetPair，$\Gamma'_{21}+\Gamma_{22} \vdash let_{q_0}(x^{r},y)\ be\ a'\ in\ b:B\{a'/z\}$。现在，根据引理 D.1，$a'\{\Delta'2\} =_\beta a\{\Delta_2\}$，其中 $\Delta'_2=\lfloor\Gamma_{21}\rfloor$。于是，$B\{a/z\}\{\Delta_2\} =B\{\Delta_2\}\{a\{\Delta_2\}/z\} =B\{\Delta'2\}\{a\{\Delta_2\}/z\} =_\beta B\{\Delta'2\}\{a'\{\Delta'2\}/z\} =B\{a'/z\}\{\Delta'2\}$。根据 PTS-DefConv 规则，$\Gamma'_{21}+\Gamma_{22} \vdash let_{q_0}(x^{r},y)\ be\ a'\ in\ b:B\{a/z\}$。否则，a 是一个值。根据反演，$a=(a_1{}^{r},a2)$。此外，存在 $\exists\Gamma_{211},\Gamma_{212}$ 使得 $\Gamma_{211} \vdash a1:^{q\cdot_{q_0}r}A1$，且 $\Gamma_{212} \vdash a2:^{q\cdot q_0}A2\{a1/x\}$，并且 $\Gamma_{21}=\Gamma_{211}+\Gamma_{212}$。于是，

– [H]令q_0(x^r, y) 为 ($a_1{}^r, a_2$) 中 b $\Longrightarrow_S^q$ [H, x $\overset{q\cdot q_0\cdot r}{\mapsto}$ a_1, y $\overset{q\cdot q_0}{\mapsto}$ a_2]b（假设 x, y 为新鲜变量）–H, x $\overset{q\cdot q_0\cdot r}{\mapsto}$ a_1, y $\overset{q\cdot q_0}{\mapsto}$ a_2 ⊨($\Gamma_1 + \Gamma_{22}$), x = a_1 :$^{q\cdot q_0\cdot r}$ A_1, y = a_2 :$^{q\cdot q_0}$ A_2 – Γ_{22}, x = a_1 :$^{q\cdot q_0\cdot r}$ A_1, y = a_2 :$^{q\cdot q_0}$ A_2 ⊢ b :^{q}B {($a_1{}^r$, a_2)/z }。

• 规则 PTS-LetUnit。与规则 PTS-LetPair类似。

•PTS-Case 规则。设: $\Gamma_{21} + \Gamma_{22}$ ⊢ case$_{q_0}$ a of x1.b1 ; x2.b2 :^{q}B {a/z } where Δ_2, z : $A_1 + A_2$ ⊢$_0$ B : s and Γ_{21} ⊢ a :$^{q\cdot q_0}$ A_1 +A2 and Γ_{22}, x1 :$^{q\cdot q_0}$ A_1 ⊢b1 :^{q}B {inj1 x1/z } and Γ_{22}, x2 :$^{q\cdot q_0}$ A_2 ⊢b2 :^{q}B {inj2 x2/z } and $\Delta_2 = \lfloor\Gamma_{21}\rfloor$.

此外, H ⊨ $\Gamma_1 + (\Gamma_{21} + \Gamma_{22})$。

若 a steps，则此 case 表达式可由归纳假设推出。

否则，a 是一个值。通过反演，a = inj$_1$ a_1 或 a = inj$_2$ a_2。

设 a = inj$_1$ a_1。那么，Γ_{21} ⊢ a_1 :$^{q\cdot q_0}$ A_1。现在，– [H]case 表达式$_{q_0}$(inj$_1$ a_1) of x_1.b_1; x_2.b_2 $\Longrightarrow_S^q$ [H , x_1 $\overset{q\cdot q_0}{\mapsto}$ a_1]b_1 （假设 x_1 是新鲜的）–H , x_1 $\overset{q\cdot q_0}{\mapsto}$ a_1 ⊨($\Gamma_1 + \Gamma_{22}$), x_1 = a_1 :$^{q\cdot q_0}$ A_1 – Γ_{22}, x_1 = a_1 :$^{q\cdot q_0}$ A_1 ⊢ b_1 :^{q}B {inj$_1$ a_1/z } 当 a = inj$_2$ a_2 时，证明类似。

□

定理 F.4（可靠性（定理 5.4）。若H ⊨ Γ 且 $\Gamma \vdash a :\ ^q A$ 且 $q \neq 0$, 则 a 要么是一个值, 要么存在 H，Γ'、a'，使得 $[H]a \Longrightarrow_S^q [H,]a'$，并且 H，⊨ Γ' 且 $\Gamma' \vdash a' :\ ^q A$。

证明。 使用引理 F.3 与 $\Gamma_1 := 0\cdot\Gamma$ 以及 $\Gamma_2 := \Gamma$。 □

推论 F.5（推论 5.5）。在 *LDC($Q_{\mathbb{N}}$)* 中：若 $\emptyset \vdash f :\ ^1 \Pi x :\ ^0 s$。$x$ 且 $\emptyset \vdash_0 A : s$，则 f A^0必定发散。

证明。为说明这一点，假设 f A^0 终止，以导出矛盾。

令 A_1 := 单元类型且 A_2 := 布尔值。

然后，对于某些j、H 和 b，我们有， [∅]f $A_1{}^0$ $\Longrightarrow_j^1$ [H](λ^0x .b) $A_1{}^0$ 以及 [∅]f $A_2{}^0$ $\Longrightarrow_j^1$[H](λ^0x .b) $A_2{}^0$。

再次， [H](λ^0x .b) $A_1{}^0$ $\Longrightarrow_S^1$ [H ,x $\overset{0}{\mapsto}$ A_1]b，以及 [H](λ^0x .b) $A_2{}^0$ $\Longrightarrow_S^1$ [H ,x $\overset{0}{\mapsto}$ A_2]b（例如，x 新鲜）。

现在，如果 [H,x $\overset{0}{\mapsto}$ A_1]b $\Longrightarrow_k^1$ [H′,x $\overset{0}{\mapsto}$ A_1, H″]b′，那么 [H,x $\overset{0}{\mapsto}$ A_2]b $\Longrightarrow_k^1$ [H′,x $\overset{0}{\mapsto}$ A_2,H″]b′，对于任意 k（根据引理 D.3）。

归约停止时，b′ 是一个值。根据可靠性，b′ 必须同时是类型 A_1和 A_2 的值，这是一种矛盾。（∵ 不存在同时具有Unit 和布尔值类型的值。）因此f$A_1{}^0$, f $A_2{}^0$, , 所以对于任意 A，f A^0, 必定发散。

由此可推得，在强规范化 PTS 中，不存在这样的 f。

使用与上文相同的论证，我们还可以证明，如果 ∅ ⊢ f :^{L} Πx :^{H}s.x 且 ∅ ⊢ A :^{H} s，那么 f A^H 必定发散。 □

推论 F.6（推论5.6）。 在 *LDC($Q_{\mathbb{N}}$)* 中：在强规范化 *PTS* 中，若 $\emptyset \vdash f :^1 \Pi x :^0 s.\Pi y :^1 x.x$且 $\emptyset \vdash_0 A : s$ 且 $\emptyset \vdash a :^1 A$，则 $f A^0\, a^1 =_\beta a$。

证明。要理解原因，我们来看看 f A^0 a^1 是如何归约的。

设 A_1 := 单元类型，且 A_2 := 布尔值。此外，设 a_1 := unit，且 a_2 := 真值。

于是，对于某个 j、H_1 和 b，我们有： [∅]f $A_1{}^0 a_1{}^1$ $\Longrightarrow_j^1$ [H_1](λ^0x 。b)$A_1{}^0 a_1{}^1$，并且 [∅] f $A_2{}^0 a_2{}^1$ $\Longrightarrow_j^1$[H_1](λ^0x 。b) $A_2{}^0$ $a_2{}^1$。

现在， $[H_1](\lambda^0 x。b)\ A_1{}^0 a_1 \Longrightarrow_S^1 [H_1, x \overset{0}{\mapsto} A_1] b\ a_1{}^1$，并且 $[H_1](\lambda^0 x。b)\ A_2{}^0 a_2{}^1 \Longrightarrow_S^1 [H_1, x \overset{0}{\mapsto} A_2] b\ a_2{}^1$ （设 x *fresh*）。

那么，对于某个 k（由引理 D.3），$[H_1, x \overset{0}{\mapsto} A_1] b\ a_1{}^1 \Longrightarrow_k^1 [H'1, x \overset{0}{\mapsto} A_1, H_2](\lambda^1 y.c)\ a_1{}^1$ 且 $[H_1, x \overset{0}{\mapsto} A_2] b\ a_2{}^1 \Longrightarrow_k^1 [H'1, x \overset{0}{\mapsto} A_2, H_2](\lambda^1 y.c)\ a_2{}^1$，。

接下来，$[H\prime 1, x \overset{0}{\mapsto} A_1, H_2](\lambda^1 y.c)\ a_1{}^1 \Longrightarrow_S^1 [H\prime 1, x \overset{0}{\mapsto} A_1, H_2, y \overset{1}{\mapsto} a_1] c$，以及 $[H\prime 1, x \overset{0}{\mapsto} A_2, H_2](\lambda^1 y.c)\ a_2{}^1 \Longrightarrow_S^1 [H\prime 1, x \overset{0}{\mapsto} A_2, H_2, y \overset{1}{\mapsto} a_2] c$（设 y 为新鲜变量）。

由于归约会终止，因此会查找 y 的值。在查找之前，这两个归约彼此无法区分。

令 $[H\prime 1, x \overset{0}{\mapsto} A_1, H_2, y \overset{1}{\mapsto} a_1] c \Longrightarrow_{k'}^1 [H\prime\prime 1, x \overset{0}{\mapsto} A_1, H\prime 2, y \overset{1}{\mapsto} a_1, H_3] c'$ 是查找 y的位置。分析步进规则可知，$c' = y$ 或 c' 是以 y 为首的适当路径（路径是一系列以变量为首的嵌套消去形式）。根据可靠性，c' 类型正确。但是，不存在既能以 a_1 为首又能以 a_2 为首的类型正确的适当路径，因为 $\emptyset \vdash a_1 :$ 单元类型，而 $\emptyset \vdash a_2 :$ 布尔值。因此，$c' = y$。

这意味着，对于任意 A 和 a，$f\ A_1{}^0\ a_1{}^1$ 归约为 a_1，$f\ A_2{}^0\ a_2{}^1$ 归约为 a_2，，因此 $f\ A^0\ a^1$ 归约为 a。因此，$f\ A^0\ a^1 =_\beta a$。

□

G LDC 的 PTS 版本中的依赖分析

引理 G.1（乘法）。 如果 $\Gamma \vdash a :^{\ell} A$，那么 $r_0 \sqcup \Gamma \vdash a :^{r_0 \sqcup \ell} A$。

证明。 对 $\Gamma \vdash a : {}^{\ell} A$进行归纳。请参见引理 C.1 的证明。 □

引理 G.2（拆分）。 若 $\Gamma \vdash a :^{\ell_1 \sqcap \ell_2} A$，则存在 Γ_1 和 Γ_2，使得 $\Gamma_1 \vdash a :^{\ell_1} A$ 和$\Gamma_2 \vdash a :^{\ell_2} A$ 和 $\Gamma = \Gamma_1 \sqcap \Gamma_2$。

证明。 已知： $\Gamma \vdash a : {}^{\ell_1 \sqcap \ell_2} A$。根据规则 PTS-SubRD， $\Gamma \vdash a : {}^{\ell_1} A$ 以及 $\Gamma \vdash a : {}^{\ell_2} A$。令 $\Gamma_1 : = \Gamma$ 以及 $\Gamma_2 : = \Gamma$，即可得出引理。 □

引理 G.3（弱化）。若 Γ_1， $\Gamma_2 \vdash a :^{\ell} A$ 且 $\Delta_1 \vdash_0 C : s$ 且 $\lfloor \Gamma_1 \rfloor = \Delta_1$，则 Γ_1， $z :^{\top} C\ \Gamma_2 \vdash a :^{\ell} A$。

证明。对 Γ_1， $\Gamma_2 \vdash a :^{\ell} A$ 进行归纳。 □

引理 G.4（替换（引理 5.1））。若 $\Gamma_1, z :^{m_0} C\ \Gamma_2 \vdash a :^{\ell} A$ 且 $\Gamma \vdash c :^{m_0} C$ 且$\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$，则 $\Gamma_1 \sqcap \Gamma, \Gamma_2\{c/z\} \vdash a\{c/z\} :^{\ell} A\{c/z\}$。

证明。 对 Γ_1， z： ${}^{m_0} C\ \Gamma_2 \vdash a : {}^{\ell} A$ 进行归纳。参见引理 C.4。 □

Theo引理 G.5（保持性（定理 5.2））。若 $\Gamma \vdash a : {}^{\ell} A$ 且 $\vdash a \rightsquigarrow a'$，则 $\Gamma \vdash a' : {}^{\ell} A$.

证明。 通过对 $\Gamma \vdash a : {}^{\ell} A$ 并对$\vdash a \rightsquigarrow a'$。遵循定理 E.6。

□

定理 G.6（进展（定理5.3））。如果 $\emptyset \vdash a : {}^{\ell} A$，那么要么 a 是一个值，要么存在a'使得 $\vdash a \rightsquigarrow a'$。

证明。 通过对 $\emptyset \vdash a : {}^{\ell} A$。请参见定理 E.7。

□

引理 G.7。 若 $H \models \Gamma_1 \sqcap \Gamma_2$ 且 $\Gamma_2 \vdash a :^{\ell} A$ 且 $l \neq \top$，则 a 要么是一个值，要么存在H'，Γ_2'，a' 使得：

- $[H]a \Longrightarrow_S^{\ell} [H']a'$

- H' $\models \Gamma_1 \sqcap \Gamma_2$
- $\Gamma_2' \vdash a' :^{\ell} A$

证明。 对 $\Gamma_2 \vdash a :^{\ell} A$ 进行归纳。参见引理 F.3。 □

定理 G.8（可靠性（定理 5.4））。如果 $H \models \Gamma$ 和 $\Gamma \vdash a :^{\ell} A$ 且 $l \neq \top$，那么要么 a 是一个值，要么存在 H'，Γ',a'，使得 $[H]a \Longrightarrow_S^{\ell} [H',]a'$ 并且 H'，$\models \Gamma'$ 且 $\Gamma' \vdash a' :^{\ell} A$。

证明。 应用引理 G.7，其中 $\Gamma_1 := \top \sqcup \Gamma$ 和 $\Gamma_2 := \Gamma$。 □

H 无限制使用

引理 H.1（乘法）。 若 $\Gamma \vdash a :^{q} A$，则 $r_0 \cdot \Gamma \vdash a :^{r_0 \cdot q} A$。

证明。 对 $\Gamma \vdash a :^{q} A$ 进行归纳。除了规则 PTS-LamOmega 之外，所有情况都与引理E.1中的情况类似。

- 规则 PTS-LamOmega。设：$q0 \cdot \Gamma \vdash \lambda^{r} x{:}A.b :^{q_0 \cdot q} \Pi x :^{r} A.B$，其中 Γ、$x{:}^{q \cdot r} A \vdash b :^{q} B$且 $q = \omega \Rightarrow r = \omega$以及 $q_0 \neq 0$。需要证明：$r0 \cdot q0 \cdot \Gamma \vdash \lambda^{r} x{:}\, A.b :^{r_0 \cdot {}_{q_0} \cdot q} \Pi x :^{r} A.B$。需要考虑两种 case 表达式：$-r_0 = 0$。根据归纳假设，$0 \cdot \Gamma$、$x :^{0} A \vdash b :^{0} B$。因此，该 case表达式可由规则 PTS-LamOmega 得出。$-r_0 \neq 0$。因此，$r0 \cdot q_0 \neq 0$。该 case 表达式可由规则 PTS-LamOmega 得出。

□

引理 H.2（独立性）。若 Γ_1，$z :^{r_0} C$ $\Gamma_2 \vdash a :^{q} A$ 且 $\neg(r_0 \ll: q)$，则 Γ_1，$z :^{0} C$ $\Gamma_2 \vdash a :^{q} A$。此处，$r_0 \ll: q$ $\exists q_0, r_0 = q + q_0$。

证明。对 Γ_1，$z :^{r_0} C$ $\Gamma_2 \vdash a :^{q} A$ 进行归纳。

- 规则 PTS-Var。有 $0 \cdot \Gamma$，$x :^{q} A \vdash x :^{q} A$ 其中 $\Delta \vdash_0 A : s$ 且 $\Delta = \lfloor \Gamma \rfloor$。此情形显然成立，因为 $q \ll: q$，且其他等级均为 0。
- 规则 PTS-Pi。有：$\Gamma_{11} + \Gamma_{21}, z :^{r_{01} + r_{02}} C$ $\Gamma_{12} + \Gamma_{22} \vdash \Pi x :^{r} A.B :^{q} s3$，其中 $\Gamma_{11}, z :^{r_{01}} C$ $\Gamma_{12} \vdash A :^{q} s1$ 且 $\Gamma_{21}, z :^{r_{02}} C$ $\Gamma_{22}, x :^{r'} A \vdash B :^{q} s2$，并且 R(s1, s2, s3) 且 $\neg(r_{01} + r_{02} \ll: q)$。现在，若 $r_{01} \ll: q$ 或 $r_{02} \ll: q$，则 $r_{01} + r_{02} \ll: q$，这与假设矛盾。因此，$\neg(r_{01} \ll: q)$ 且 $\neg(r_{02} \ll: q)$。于是，根据归纳假设和规则 PTS-Pi，此情形成立。• 规则 PTS-LamOmega。有：$q0 \cdot \Gamma_1, z :^{q_0 \cdot r_0} C$, $q0 \cdot \Gamma_2 \vdash \lambda^{r} x : A.b :^{q_0 \cdot q} \Pi x :^{r} A.B$，其中 $\Gamma_1, z :^{r_0} C$ $\Gamma_2, x :^{q \cdot r} A \vdash b :^{q} B$，且 $q = \omega \Rightarrow r = \omega$且 $q_0 \neq 0$且 $\neg(q0 \cdot r_0 \ll: q0 \cdot q)$。若 $r_0 \ll: q$，则 $q0 \cdot r_0 \ll: q0 \cdot q$，这与假设矛盾。因此，$\neg(r_0 \ll: q)$。于是，根据归纳假设和规则 PTS-LamOmega，此情形成立。
- PTS-应用规则。已知：$\Gamma_{11} + \Gamma_{21}, z :^{r_{01} + r_{02}} C$ $\Gamma_{12} + \Gamma_{22} \vdash b\ a :^{r} B\{a/x\}$，其中 $\Gamma_{11}, z :^{r_{01}} C$ $\Gamma_{12} \vdash b :^{q} \Pi x :^{r} A.B$，且 $\Gamma_{21}, z :^{r_{02}} C$ $\Gamma_{22} \vdash a :^{q \cdot r} A$，以及 $\neg(r_{01} + r_{02} \ll: q)$。有两种情况需要考虑。$-r = 0$。那么，$\Gamma_{21}, z :^{0} C$ $\Gamma_{22} \vdash a :^{0} A$。现在，若 $r_{01} \ll: q$，则 $r_{01} + r_{02} \ll: q$，这与矛盾相矛盾。因此，$\neg(r_{01} \ll: q)$。那么，这种情况可由归纳假设和 PTS-App规则得出。$-r \neq 0$。那么，$r = 1 + r'$，其中 r' 为某个项。因此，$\neg(r_{01} \ll: q)$ 且 $\neg(r_{02} \ll: q \cdot r)$。那么，这种情况可由归纳假设和 PTS-App规则得出。

• PTS-Pair规则。已有： $\Gamma_{11}+\Gamma_{21}$，z:$^{r_{01}+r_{02}}$C $\Gamma_{12}+\Gamma_{22}\vdash$(a$_1{}^r$, a2):^{q} Σx :^{r}A1.A2，其中 Γ_{11}，z :$^{r_{01}}$C $\Gamma_{12}\vdash$a1:$^{q\cdot r}$A1且 Γ_{21}，z:$^{r_{02}}$C $\Gamma_{22}\vdash$a2:^{q}A2{a1/x}，以及 $\neg(r_{01}+r_{02}\ll: q)$。这里同样有两种需要考虑的情况。$-r=0$。于是， Γ_{11}，z:^{0}C $\Gamma_{12}\vdash$a1 :^{0}A1。现在，如果 $r_{02}\ll: q$，那么 $r_{01}+r_{02}\ll: q$，这就导致矛盾。因此， $\neg(r_{02}\ll: q)$。因此，这个 case 表达式可由归纳假设和 PTS-Pair规则得出。$-r\neq 0$。于是，对于某个 r'，有 $r=1+r'$。因此， $\neg(r_{01}\ll: q\cdot r)$且 $\neg(r_{02}\ll: q)$。因此，这个 case 表达式可由归纳假设和 PTS-Pair规则得出。

• PTS-LetPair 规则。已知： $\Gamma_{11}+\Gamma_{21}$, z :$^{r_{01}+r_{02}}$C $\Gamma_{12}+\Gamma_{22}\vdash$let$_{q_0}$(x ^{r}, y) be a in b :^{q}B {a/z }，其中 Δ, z : Σx :^{r}A1.A$_2\vdash_0$B : s，且 Γ_{11}, z :$^{r_{01}}$C $\Gamma_{12}\vdash$a :$^{q\cdot q_0}$ Σx :^{r}A1.A2，且 Γ_{21}, z :$^{r_{02}}$C Γ_{22}, x :$^{q\cdot_{q_0} r}$A1, y :$^{q\cdot q_0}$ A$_2\vdash$b :^{q}B {(x ^{r}, y)/z }，并且 $q_0<: 1$ 且 $\neg(r_{01}+r_{02}\ll: q)$。由于 $q_0<: 1$，因此 $q_0=1+q'0$，对某个 $q'0$ 成立。因此， $\neg(r_{01}\ll: q\cdot q0)$ 且 $\neg(r_{02}\ll: q)$。于是，此情形由归纳假设和 PTS-LetPair 规则得证。

• 其他情况同理。

□

引理 H.3（因子分解）。如果 $\Gamma\vdash a:^q A$ 且 $q\neq 0$，则存在 Γ' 使得 $\Gamma'\vdash a:^1 A$且 $\Gamma<: q\cdot\Gamma'$。

证明。对 $\Gamma\vdash$a :^{q}A 做归纳。

• 规则 PTS-Pi。已有： $\Gamma_1+\Gamma_2\vdash\Pi$x :^{r}A.B :^{q}s3，其中 $\Gamma_1\vdash$A :^{q}s1 且 Γ_2，x :^{r_0}A $\vdash$B :^{q} s2。根据归纳假设， $\exists\Gamma'1,\Gamma'2$ 使得 $\Gamma'_1\vdash$A :^{1} s1 且 $\Gamma'2$，x :$^{r'}$A $\vdash$B :^{1} s2，其中 $\Gamma_1<:$ q$\cdot\ \Gamma'1$ 且$\Gamma_2<:$ q$\cdot\Gamma'2$。于是，该情况由规则 PTS-Pi 得出。• 规则 PTS-LamOmega。已有： q0$\cdot\ \Gamma\vdash\lambda^r$x : A.b :$^{q_0\cdot q}\Pi$x :^{r}A.B，其中 Γ，x :$^{q\cdot r}$A $\vdash$b :^{q}B 且 $q=\omega\Rightarrow r=\omega$ 和 $q_0\neq 0$。需要考虑两种情况：$-q=\omega$。于是 q0$\cdot$ q $=\omega$。这里，我们需要定义一个上下文上的运算。对于一个上下文 Γ，定义 Γ{0,ω} 为：

$$\emptyset^{\{0,\omega\}}=\emptyset$$

$$(\Gamma, x:^q A)^{\{0,\omega\}}=\begin{cases}\Gamma^{\{0,\omega\}}, x:^q A \text{ if } q\in\{0,\omega\}\\ \Gamma^{\{0,\omega\}}, x:^0 A \text{ otherwise}\end{cases}$$

现在，由于 $q_0\cdot\ \Gamma\vdash\lambda^r$x : A.b :$^\omega\ \Pi$x :^{r}A.B，根据引理 H.2，$(q_0\cdot\ \Gamma)^{\{0,\omega\}}\vdash\lambda^r$x : A.b :$^\omega\Pi$x :^{r} A.B 。

根据规则 PTS-SubR，$(q_0\cdot\ \Gamma)^{\{0\omega\}}\vdash\lambda^r$x:A.b : Πx :^{r} A.B.

接下来，$q_0\cdot\ \Gamma<:(q_0\cdot\ \Gamma)^{\{0,\omega\}}=\omega\cdot(q_0\cdot\ \Gamma)^{\{0,\omega\}}=q_0\cdot q\cdot(q_0\cdot\ \Gamma)^{\{0,\omega\}}$。

注意，对于 $\mathbb{N}^\omega_=$和 Q_{Lin} 的情形，我们有，$q_0\cdot\ \Gamma=(q_0\cdot\ \Gamma)^{\{0,\omega\}}$。

$-q\neq\omega$。根据归纳假设，Γ'，x :$^{r'}$A $\vdash$b :^{1}B，其中 $\Gamma<: q\cdot\Gamma'$和 $q\cdot r<: q\cdot r'$。

由于 $q\cdot r<: q\cdot r'$，且 $q\notin\{0,\omega\}$，因此 $r<: r'$。

根据规则 PTS-SubL，Γ'，$x:{}^rA\vdash b:{}^1B$。

因此，这种情况由 b 得出 y 规则 PTS-LamOmega。

• 规则 PTS-应用。已知： $\Gamma_1+\Gamma_2\vdash b\,a^r:{}^qB\{a/x\}$其中 $\Gamma_1\vdash b:{}^q\Pi x:{}^rA$。$B$且 $\Gamma_2\vdash a:{}^q$ ${}^rA$. 有两种情况需要考虑：

$-r=0$。根据归纳假设，$\exists\Gamma'_1$ 使得 $\Gamma'_1\vdash b:^1\Pi x:^rA$。$B$ 以及 $\Gamma_1<:q\cdot$ Γ'_1。接下来，根据乘法引理，$0\cdot\ \Gamma_2\vdash a:^0A$。因此，此 case 表达式由规则PTS-App得出。

– $r \neq 0$。根据归纳假设，$\exists \Gamma_1', \Gamma_2'$ 使得 $\Gamma_1' \vdash b :^1 \Pi x :^r A.B$ 以及 $\Gamma_2' \vdash a :^1 A$，其中 $\Gamma_1 <: q \cdot \Gamma_1'$ 且 $\Gamma_2 <: q \cdot r \cdot \Gamma_2'$。然后，根据乘法引理，$r \cdot \Gamma_2' \vdash a :^r A$。因此，此 case 表达式由 PTS-App规则得出。

- 规则 PTS-Pair。已知：$\Gamma_1 + \Gamma_2 \vdash (a_1{}^r, a_2) :^q \Sigma x :^r A_1.A_2$ 其中 $\Gamma_1 \vdash a_1 :^q A_1$和$\Gamma_2 \vdash a_2 :^q A_2\{a_1/x\}$.

  有两种情况需要考虑。
  – $r = 0$。然后，根据乘法引理，$0 \cdot \Gamma_1 \vdash a_1 :^0 A_1$。根据归纳假设，$\exists \Gamma_2'$ 使得 $\Gamma_2' \vdash a_2 :^1 A_2\{a_1/x\}$ 且 $\Gamma_2 <: q \cdot \Gamma_2'$。因此，此情形由规则 PTS-Pair得出。
  – $r \neq 0$。根据归纳假设，$\exists \Gamma_1'$、Γ_2'使得 $\Gamma_1' \vdash a_1 :^1 A_1$ 且 $\Gamma_2' \vdash a_2 :^1 A_2\{a_1/x\}$，以及 $\Gamma_1 <: (q \cdot r) \cdot \Gamma_1'$和 $\Gamma_2 <: q \cdot \Gamma_2'$。然后，根据乘法引理，$r \cdot \Gamma_1' \vdash a_1 :^r A_1$。因此，该 case 表达式可由 PTS-Pair规则推出。

- 规则 PTS-LetPair。已知：$\Gamma_1 + \Gamma_2 \vdash \text{let}_{q_0}(x^r, y)$ be a in b $:^q B\{a/z\}$ where $\Delta, z : \Sigma x :^r A_1.A_2 \vdash_0 B : s$ and $\Gamma_1 \vdash a :^{q \cdot q_0} \Sigma x :^r A_1.A_2$ and $\Gamma_2, x :^{q \cdot q_0 \cdot r} A_1, y :^{q \cdot q_0} A_2 \vdash b :^q B\{(x^r, y)/z\}$.

  需要考虑两种 case 表达式：
  – $q = \omega$。现在，根据引理 H.2，$(\Gamma_1 + \Gamma_2)\{0,\omega\} \vdash \text{let}_{q_0}(x^r, y)$ be a in b $:^\omega B\{a/z\}$。根据 PTS-SubR规则，$(\Gamma_1 + \Gamma_2)^{\{0,\omega\}} \vdash \text{let}_{q_0}(x^r, y)$ be a in b $:^1 B\{a/z\}$。接下来，$\Gamma_1 + \Gamma_2 <: (\Gamma_1 + \Gamma_2)\{0,\omega\} = \omega \cdot (\Gamma_1 + \Gamma_2)^{\{0,\omega\}}$。注意，在 case 表达式 $\mathbb{N}_=^\omega$和 $\mathrm{Q}_{\mathrm{Lin}}$ 中，我们有 $\Gamma_1 + \Gamma_2 = (\Gamma_1 + \Gamma_2)^{\{0,\omega\}}$。– $q \neq \omega$。根据归纳假设，$\Gamma_1' \vdash a :^1 \Sigma x :^r A_1.A_2$和 $\Gamma_2', x :^{r_1'} A_1, y :^{r_2'} A_2 \vdash b :^1 B\{(x^r, y)/z\}$，其中$\Gamma_1 <: q \cdot q_0 \cdot \Gamma_1'$和 $\Gamma_2 <: q \cdot \Gamma_2'$和 $q \cdot q_0 \cdot r <: q \cdot r_1'$和 $q \cdot q_0 <: q \cdot r_2'$。现在，由于 $q \notin \{0, \omega\}$，所以 $q_0 \cdot r <: r'1$ 且 $q_0 <: r'2$。因此，根据 PTS-SubL规则，$\Gamma_2', x :^{q_0 \cdot r} A_1, y :^{q_0} A_2 \vdash b :^1 B\{(x^r, y)/z\}$。接下来，根据乘法引理，$q_0 \cdot \Gamma_1' \vdash a :^{q_0} \Sigma x :^r A_1.A_2$。于是，此情形由 PTS-LetPair 规则得证。

- 其他情况的证明类似。

□

引理 H.4（拆分）。若 $\Gamma \vdash a :^{q_1+q_2} A$，则存在 Γ_1 和 Γ_2，使得 $\Gamma_1 \vdash a :^{q_1} A$ 且$\Gamma_2 \vdash a :^{q_2} A$ 以及 $\Gamma = \Gamma_1 + \Gamma_2$。

证明。若 $q_1 + q_2 = 0$，则 $\Gamma_1 := 0 \cdot \Gamma$ 且 $\Gamma_2 := \Gamma$。
否则，根据引理H.3，$\exists \Gamma'$ 使得 $\Gamma' \vdash a :^1 A$ 且 $\Gamma <: (q_1 + q_2) \cdot \Gamma'$。
于是，对于某个 Γ_0，有 $\Gamma = (q_1 + q_2) \cdot \Gamma' + \Gamma_0$。对于 $\mathbb{N}_=^\omega$ 和 $\mathrm{Q}_{\mathrm{Lin}}$ 的情况，我们选择 Γ_0 使得 $\Gamma_0 = \Gamma^{\{0,\omega\}}{}_0$。现在，根据引理H.1，$q_1 \cdot \Gamma' \vdash a :^{q_1} A$ 和 $q_2 \cdot \Gamma' \vdash a :^{q_2} A$。
接下来，$q_2 \cdot \Gamma' + \Gamma_0 <: q_2 \cdot \Gamma'$。于是，根据规则 PTS-SubL，$q_2 \cdot \Gamma' + \Gamma_0 \vdash a :^{q_2} A$。将 $\Gamma_1 := q_1 \cdot \Gamma'$ 和 $\Gamma_2 := q_2 \cdot \Gamma' + \Gamma_0$，即可得出该引理。 □

引理 H.5（弱化）。若 $\Gamma_1, \Gamma_2 \vdash a :^q A$ 且 $\Delta_1 \vdash_0 C : s$ 且 $\lfloor \Gamma_1 \rfloor = \Delta_1$，则 $\Gamma_1, z :^0 C\ \Gamma_2 \vdash a :^q A$。

证明。对 $\Gamma_1, \Gamma_2 \vdash a :^q A$ 进行归纳。 □

引理 H.6（替换）。若 $\Gamma_1, z :^{r_0} C\ \Gamma_2 \vdash a :^q A$ 且 $\Gamma \vdash c :^{r_0} C$ 且 $\lfloor \Gamma_1 \rfloor = \lfloor \Gamma \rfloor$，则$\Gamma_1 + \Gamma, \Gamma_2\{c/z\} \vdash a\{c/z\} :^q A\{c/z\}$。

证明。对 Γ_1进行归纳，$z :^{r_0} C\ \Gamma_2 \vdash a :^q A$。除规则 PTS-LamOmega之外的所有 case 表达式都与引理E.5中的情况类似。

• 规则PTS-LamOmega。 有： $q0 \cdot \Gamma_1$， $z:^{q_0 \cdot r_0} C$， $q0 \cdot \Gamma_2 \vdash \lambda^r x{:}A.b :^{q_0 \cdot q} \Pi x:^r A.B$，其中 Γ_1， $z:^{r_0} C\ \Gamma_2$， $x :^{q \cdot r} A \vdash b :^q B$，以及$q = \omega \Rightarrow r = \omega$和 $q_0 \neq 0$。此外， $\Gamma \vdash c:^{q_0 \cdot r_0} C$，其中 $\lfloor\Gamma\rfloor = \lfloor\Gamma_1\rfloor$。需要考虑两种情况： $-r_0 = 0$。那么， $0 \cdot \Gamma \vdash c :^0 C$。根据归纳假设， Γ_1, Γ_2 $\{c/z\}$， $x :^{q \cdot r} A\{c/z\} \vdash b\{c/z\} :^q B\{c/z\}$。因此，此情况由规则PTS-LamOmega 和 PTS-SubL得出。 $-r_0 \neq 0$。那么$q0 \cdot r_0 \neq 0$。根据因子分解引理，存在 $\exists\Gamma'$，使得 $\Gamma' \vdash c :^1 C$，其中 $\Gamma <: q0 \cdot r0 \cdot \Gamma'$。根据乘法引理， $r0 \cdot \Gamma' \vdash c:^{r_0} C$。根据归纳假设， $\Gamma_1 + r0 \cdot \Gamma' \Gamma_2$ $\{c/z\}$， $x :^{q \cdot r} A\{c/z\} \vdash b\{c/z\} :^q B\{c/z\}$。根据规则 PTS-LamOmega， $q0 \cdot \Gamma_1 + q0 \cdot r0 \cdot \Gamma'$， $q0 \cdot \Gamma_2\{c/z\} \vdash \lambda^r x{:}\ A\{c/z\}.b\{c/z\} :^{q_0 \cdot q} \Pi x :^r A\{c/z\}.B\ \{c/z\}$。因此，此情况由 PTS-SubL规则得出。

□

引理 H.7（λ 反演）。若 $\Gamma \vdash \lambda^r x : A。b :^q C$， 则 $\exists A'$、B'， Γ_0， q_0, q_1 使得：

- $\Gamma_0, x:^{q_0 \cdot r} A'\ \vdash b :^{q_0} B'$
- $\Gamma <: q_1 \cdot \Gamma_0$ 和 $q_1 \cdot q_0 <: q$
- $q_0 = \omega \Rightarrow r = \omega$ 和 $q_1 \neq 0$
- $A =_\beta A'$ 且 $C =_\beta \Pi x :^r A'.B'$ 且 $\Delta \vdash_0 C :\ s$，其中 $\Delta = \lfloor\Gamma\rfloor$。

证明。通过对 $\Gamma \vdash \lambda^r x : A。b :^q C$ 进行归纳。 □

定理 H.8（保持性）。如果 $\Gamma \vdash a :^q A$ 且 $\vdash a \rightsquigarrow a'$，那么 $\Gamma \vdash a' :^q A$。

证明。 对 $\Gamma \vdash a :^q A$ 进行归纳，并对 $\vdash a \rightsquigarrow a'$进行反演。除了PTS-App规则之外的所有情形，都与引理E.6中的情形类似。

• PTS-App规则。 有： $\Gamma_1 + \Gamma_2 \vdash b\ a^r :^q B\{a/x\}$，其中 $\Gamma_1 \vdash b :^q \Pi x :^r A.B$ 且 $\Gamma_2 \vdash a :^{q \cdot r} A$。令 $\vdash b\ a^r \rightsquigarrow c$。通过反演： $- \vdash b\ a^r \rightsquigarrow b'\ a^r$，当 $\vdash b \rightsquigarrow b'$ 时。需证明： $\Gamma_1 + \Gamma_2 \vdash b'\ a^r :^q B\{a/x\}$。根据归纳假设和 PTS-App规则可得。 $-b = \lambda^r x : A'.b'$ 且 $\vdash b\ a^r \rightsquigarrow b'\{a/x\}$。需证明： $\Gamma_1 + \Gamma_2 \vdash b'\{a/x\} :^q B\{a/x\}$。对 $\Gamma_1 \vdash \lambda^r x : A'.b' :^q \Pi x :^r A.B$ 应用引理 H.7， 可得： $* \Gamma_0, x :^{q_0 \cdot r} A''\ \vdash b' :^{q_0} B'$

  * $\Gamma_1 <: q_1 \cdot \Gamma_0$ 和 $q_1 \cdot q_0 <: q$
  * $q_0 = \omega \Rightarrow r = \omega$ 和 $q_1 \neq 0$
  * $A'' =_\beta A'$ 和 $\Pi x :^r A.B =_\beta \Pi x :^r A''.B'$。现在

  需要考虑三种情况。
  * $q_0 = 0$。由于 $q_1 \cdot q_0 <: q$，因此 $q = 0$。然后，根据替换引理和规则 PTS-SubL， $\Gamma_1 + \Gamma_2 \vdash b'\{a/x\} :^0 B\{a/x\}$。
  * $q_0 = \omega$。因此 $r = \omega$。现在，如果 $q = 0$，那么根据替换引理和规则 PTS-SubL， $\Gamma_1 + \Gamma_2 \vdash b'\{a/x\} :^0 B\{a/x\}$。如果 $q \neq 0$，那么 $q \cdot r = \omega$。因此 $\Gamma_2 \vdash a :^\omega A$。接下来， Γ_0， $x :^\omega A''\ \vdash b' :^\omega B'$。根据乘法引理和规则 PTS-SubL， Γ_1， $x :^\omega A''\ \vdash b' :^\omega B'$。

根据替换引理，$\Gamma_1 + \Gamma_2 \vdash b'\{a/x\} :^{\omega} B'\{a/x\}$。因此，这个 case 表达式根据规则 PTS-Conv 规则和PTS-SubR。

* $q_0 \notin \{0, \omega\}$。根据引理 H.3，存在 $\exists \Gamma' 0$和r'，使得 $\Gamma' 0, x :^{r'} A'' \vdash b' :^{1} B'$，且 $\Gamma_0 <: q0 \cdot \Gamma' 0$，并且 $q0 \cdot r <: q0 \cdot r'$。由于 $q_0 \notin \{0, \omega\}$，所以 $r <: r'$。因此，根据 PTS-SubL规则，$\Gamma' 0, x :^{r} A'' \vdash b' :^{1} B'$。现在，根据引理 H.1，$q \cdot \Gamma' 0, x :^{q \cdot r} A'' \vdash b' :^{q} B'$。根据PTS-SubL规则，$\Gamma_1, x :^{q \cdot r} A'' \vdash b' :^{q} B'$。于是，这种情况由 PTS-Conv 规则和替换引理得证。

□

定理 H.9（进展）。若 $\emptyset \vdash a :^{q} A$，则$a$ 是一个值，或者存在a'，使得$\vdash a \rightsquigarrow a'$。

证明。 对 $\emptyset \vdash a :^{q} A$ 作归纳。遵循定理 E.7 的证明。 □

注意： 多重代换引理和展开引理（引理 F.1 和 F.2）对 LDC($\mathcal{Q}_{\mathbb{N}}^{\omega}$) 成立。

引理 H.10。 若$H \models \Gamma_1 + \Gamma_2$ *and* $\Gamma_2 \vdash a :^{q} A$ *and* $q \neq 0$，则a 要么是一个值，要么存在H'，Γ_2'，a' 使得：

- $[H]a \Longrightarrow_S^q [H']a'$
- $H' \models \Gamma_1 + \Gamma' 2$
- $\Gamma'_2 \vdash a' :^{q} A$

证明。 对 $\Gamma_2 \vdash a :^{q} A$进行归纳。除规则PTS-应用 外，其他所有情况都与引理F.3中的情况类似。

- 规则 PTS-应用。已知：$\Gamma_{21} + \Gamma_{22} \vdash b\ a^{r} :^{q} B\{a/x\}$，其中 $\Gamma_{21} \vdash b :^{q} \Pi x :^{r} A$。$B$ 且 $\Gamma_{22} \vdash a :^{q \cdot r} A$。此外，$H \models \Gamma_1 + (\Gamma_{21} + \Gamma_{22})$。若 b 发生归约，则此情况由归纳假设得出。

否则，b 是一个值。通过反演，$b = \lambda^{r} x : A'.b'$。
利用引理 H.7关于 $\Gamma_{21} \vdash \lambda^{r} x{:}\, A'.b' :^{q} \Pi x{:}^{r} A$。$B$，我们得到：
– $\Gamma_{20}, x :^{q_0 \cdot r} A'' \vdash b' :^{q_0} B'$– $\Gamma_{21} <: q_1 \cdot \Gamma_{20}$ 且 $q_1 \cdot q_0 <: q$– $q_0 = \omega \Rightarrow r = \omega$ 且 $q_1 \neq 0$– $A''\{\Delta_2\} =_{\beta} A'\{\Delta_2\}$ 且 $\Pi x :^{r} A$。$B\{\Delta_2\} =_{\beta} \Pi x :^{r} A''.B'\{\Delta_2\}$，其中 $\Delta_2 = \lfloor \Gamma_{21} \rfloor$。

现在需要考虑两种情况。
– $q_0 = \omega$。因此 $r = \omega$。此外，由于 $q \neq 0$，所以 $q \cdot r = \omega$。因此，$\Gamma_{20}, x :^{\omega} A'' \vdash b' :^{\omega} B'$。根据乘法引理和 PTS-SubL规则，$\Gamma_{21}, x :^{\omega} A'' \vdash b' :^{\omega} B'$。然后，根据 PTS-DefConv规则，$\Gamma_{22} \vdash a :^{\omega} A''$。此外，$\Gamma_{21}, x = a :^{\omega} A'' \vdash b' :^{\omega} B'$。根据 PTS-SubR规则，$\Gamma_{21}, x = a :^{\omega} A'' \vdash b' :^{q} B'$。现在，

* $[H](\lambda^{r} x : A.b)\ a^{r} \Longrightarrow_S^q [H, x \overset{\omega}{\mapsto} a]b'$
* $H, x \overset{\omega}{\mapsto} a \models (\Gamma_1 + \Gamma_{21}), x = a :^{\omega} A''$
  * $\Gamma_{21}, x = a :^{\omega} A'' \vdash b' :^{q} B\{a/x\}$，依据规则 PTS-DefConv。– $q_0 \neq \omega$。注意，如果 $q_0 = 0$，那么 $q = 0$，这就构成矛盾。因此 $q_0 \notin \{0, \omega\}$。

由引理 H.3，$\exists \Gamma'_{20}$以及 r'，使得 $\Gamma'_{20}, x :^{r'} A'' \vdash b' :^{1} B'$且 $\Gamma_{20} <: q_0 \cdot \Gamma'_{20}$且 $q_0 \cdot r <: q_0 \cdot r'$。

由于 $q_0 \notin \{0, \omega\}$，因此 $r <: r'$。于是，根据 PTS-SubL规则，$\Gamma'_{20}, x :^{r} A'' \vdash b' :^{1} B'$。现在，根据引理 H.1，$q \cdot \Gamma'_{20}, x :^{q \cdot r} A'' \vdash b' :^{q} B'$。根据 PTS-SubL规则，$\Gamma_{21}, x :^{q \cdot r} A'' \vdash b' :^{q} B'$。

接下来，根据规则 PTS-DefConv，$\Gamma_{22} \vdash a :^{q \cdot r} A''$。然后，$\Gamma_{21}, x = a :^{q \cdot r} A'' \vdash b' :^{q} B'$。因此，

$$* [H, x \overset{q}{\mapsto} a](\lambda^{r} x{:}\ A.b) a^{r} \Longrightarrow_{S}^{q} [H, x \overset{q}{\mapsto} a] b \quad H' = H, x \overset{q \cdot r}{\mapsto} a \quad \Gamma_1 + \Gamma_{21}, x = a :^{q \cdot r} A'' \vdash b' :^{q} B' \quad * \Gamma_{21}, x = a :^{q \cdot r} A'' \vdash b' :^{q} B'$$

根据规则 PTS-DefConv。

□

定理 H.11（可靠性）。若 $H \models \Gamma$ 且 $\Gamma \vdash a :^{q} A$ 且 $q \neq 0$，则 a 是一个值，或存在 H'、Γ'、a'，使得 $[H]a \Longrightarrow_{S}^{q} [H']a'$，且 $H' \models \Gamma'$ 且 $\Gamma' \vdash a' :^{q} A$。

证明。使用引理 H.10，其中 $\Gamma_1 := 0 \cdot \Gamma$ 且 $\Gamma_2 := \Gamma$。 □

定理 H.12（定理 6.1）。*LDC($Q_{\mathbb{N}}^{\omega}$)* 满足类型可靠性和堆可靠性。

证明。由定理 H.8、H.9 和 H.11可得。 □

I LDC 与其他演算的比较

定理 I.1（定理7.1）。从 *LNL* λ-演算到 *LDC(Q_{Lin})* 的翻译如图*10*所示，具有可靠性：

- 若 $\Theta;\ \Gamma \vdash_{\mathcal{L}} e :\ A$，则 $\overline{\Theta}^{\omega}, \overline{\Gamma}^{1} \vdash e^{\circ} :^{1} \overline{A}^{\circ}$
- 如果 $\Theta \vdash_{C} t : X$，那么 $\overline{\Theta}^{\omega} \vdash \overline{t}^{\circ} :^{\omega} \overline{X}^{\circ}$
- 如果 $e =_{\beta} f$，那么 $\overline{e}^{\circ} =_{\beta} \overline{f}^{\circ}$。如果 $s =_{\beta} t$ 那么 $\overline{s}^{\circ} =_{\beta} \overline{t}^{\circ}$

证明。对类型判断中的 $\Theta; \Gamma \vdash_{\mathcal{L}} e : A$ 和 $\Theta \vdash_{C} t : X$ 作相互归纳。对用于 β-等式的 $e =_{\beta} f$ 和 $s =_{\beta} t$ 作分类讨论。 □

定理 I.2（定理7.2）。以 $Q_{\mathbb{N}}^{\omega}$ 作为参数化结构，如果 $\Gamma \vdash a :\ A$ 属于 *GrAD*，则$\Gamma \vdash a :^{1} A$ 属于 *LDC*。此外，如果 $\vdash a \rightsquigarrow a'$ 属于*GrAD*，则 $\vdash a \rightsquigarrow a'$ 属于 *LDC*。

证明：由 GraD 类型判断上的归纳法得出。 □

定理 I.3（定理7.3）。以 $\mathcal{L}$ 作为参数化结构，若 $\Gamma \vdash a :\ ^{\ell} A$ 属于*DDC*$^{\top}$，则$\Gamma \vdash a :\ ^{\ell} A$ 属于 *LDC*。此外，若 $\vdash a \rightsquigarrow a'$ 属于*DDC*$^{\top}$，则 $\vdash a \rightsquigarrow a'$ 属于 *LDC*。

证明。对 DDC$^{\top}$ 类型判断进行归纳。 □

LDC($\mathcal{L}$) 中合并和分叉的 J 推导

命题 J.1。 $\emptyset \vdash c_1 :\ T_{\ell_1}\ T_{\ell_2}\ A \to T_{\ell_1 \sqcup \ell_2}\ A$ 其中

$$c_1 := \lambda x.\eta_{\ell_1 \sqcup \ell_2}\ (\text{let}\ (y^{\ell_2}, _)\ \text{be}\ (\text{let}\ (z^{\ell_1}, _)\ \text{be}\ x\ \text{in}\ z)\ \text{in}\ y),\ \textit{where}\ \eta_{\ell}\ a \triangleq (a^{\ell}, \text{unit})$$

.

证明。 (1) $x :\ ^{\perp}\ T_{\ell_1}\ T_{\ell_2}\ A \vdash x :\ ^{\ell_1}\ T_{\ell_1}\ T_{\ell_2}\ A$ [根据规则 ST-VarD 和 ST-SubRD]

(2) $x :^{\perp} T_{\ell_1}\ T_{\ell_2} A, z :^{\ell_1} T_{\ell_2} A \vdash z :^{\ell_1} T_{\ell_2} A$ (3) $x :^{\perp} T_{\ell_1}\ T_{\ell_2} A \vdash$ 令 $(z^{\ell_1}, _)$ 为 x，且 $z :^{\ell_1} T_{\ell_2} A$ [根据 (1) 和 (2)，应用 ST-LetPairD 规则](4) $x :^{\perp} T_{\ell_1}\ T_{\ell_2} A \vdash$ 令 $(z^{\ell_1}, _)$ 为 x，且 $z :^{\ell_1 \sqcup \ell_2} T_{\ell_2} A$ [根据 (3)，应用 ST-SubRD 规则](5) $x :^{\perp} T_{\ell_1}\ T_{\ell_2} A, y :^{\ell_1 \sqcup \ell_2} A \vdash y :^{\ell_1 \sqcup \ell_2} A$

(6) $x:{}^{\perp}\ T_{\ell_1}\ T_{\ell_2} A \vdash$ 令 $(y^{\ell_2},\ _)$ 为 (令 $(z^{\ell_1},\ _)$ 为 x 于 z) 于 $y:{}^{\ell_1 \sqcup \ell_2} A$ [由 (4) 和 (5)，根据规则 ST-LetPairD]。

(7) x :$^{\perp}$ T_{ℓ_1} T_{ℓ_2}A ⊢ $\eta_{\ell_1 \sqcup \ell_2}$(令 (y$^{\ell_2}$,_) 为 (令 (z$^{\ell_1}$,_) 为 x，结果为 z)，结果为 y) :$^{\perp}$ $T_{\ell_1 \sqcup \ell_2}$A

□

命题 J.2。 $\emptyset \vdash c_2:\ T_{\ell_1 \sqcup \ell_2} A \to T_{\ell_1}\ T_{\ell_2} A$ 其中 $c_2 := \lambda x.\eta_{\ell_1}\ \eta_{\ell_2}$(令 (y$^{\ell_1 \sqcup \ell_2}$,_) 为 x 中的 y)，其中 $\eta_{\ell} a \triangleq (a^{\ell}$，单元类型)

.

证明。 (1) $x:{}^{\perp}\ T_{\ell_1 \sqcup \ell_2} A \vdash x:{}^{\ell_1 \sqcup \ell_2}\ T_{\ell_1 \sqcup \ell_2} A$ [根据规则 ST-VarD 和 ST-SubRD]

(2) x:$^{\perp}$ $T_{\ell_1 \sqcup \ell_2}$A,y:$^{\ell_1 \sqcup \ell_2}$A ⊢ y:$^{\ell_1 \sqcup \ell_2}$A (3) x :$^{\perp}$ $T_{\ell_1 \sqcup \ell_2}$A ⊢ let(y$^{\ell_1 \sqcup \ell_2}$,_)be x in y :$^{\ell_1 \sqcup \ell_2}$A [由 (1) 和 (2)，根据ST-LetPairD 规则](4) x :$^{\perp}$ $T_{\ell_1 \sqcup \ell_2}$A ⊢ η_{ℓ_2}(let(y$^{\ell_1 \sqcup \ell_2}$,_)bex in y) : $^{\ell_1}$ T_{ℓ_2}A (5) x :$^{\perp}$ $T_{\ell_1 \sqcup \ell_2}$A ⊢ $\eta_{\ell_1}\ \eta_{\ell_2}$(let(y$^{\ell_1 \sqcup \ell_2}$,_) be x in y) :$^{\perp}$ $T_{\ell_1}\ T_{\ell_2}$A

□