

分层指称系统 95

分层指称系统 (LRS): 形式化规范

本文档详细定义了一个分层指称系统 (LRS)，包括其语法、证明论、模型论语义，并提供了关键性质的证明。该系统的一个显著特征是其对象语言中的定义连接符 $:=$ 具有非交换性，并且其语义模型被设计为能够使一套特定的推理规则（包括一个“具体化”规则 Reif）成立。

第 1 节 语法 (Syntax) - \mathcal{L}_0

定义 1.1 (字母表)

系统的字母表 Σ 包含：

- 一个可数的符号集 $\mathcal{S} = \{s_1, s_2, \dots\}$ 。
- 一个特殊的构造符 `ref`。
- 三个特殊的原子常量：`()`、`⊥` 和 `⊥_cycle`。其中 `⊥_cycle` 用于在对象语言中表示指称循环。
- 一个二元连接符： $:=$ 。

定义 1.2 (项)

\mathcal{L}_0 的项集 \mathcal{T} 是满足以下条件的最小集合：

- 原子项: 若 $s \in \mathcal{S}$ ，则 $s \in \mathcal{T}$ 。常量 `()`、`⊥` 和 `⊥_cycle` 也是 \mathcal{T} 中的原子项。
- 定义项: 若 $t_1, t_2 \in \mathcal{T}$ ，则 $(t_1 := t_2) \in \mathcal{T}$ 。

第 2 节 证明论 (Proof Theory)

证明论由 $\Gamma \vdash t_1 \sim t_2$ 形式的判断 (Sequent) 构成，其中 Γ 是一个由该形式判断组成的集合（上下文）， \sim 是一个旨在捕获语义等价性的元语言二元关系。

定义 2.1 (推理规则)

\sim 是一个等价关系，满足以下公理：

- (Reflexivity):
$$\frac{}{\Gamma \vdash t \sim t}$$
- (Symmetry):
$$\frac{\Gamma \vdash t_1 \sim t_2}{\Gamma \vdash t_2 \sim t_1}$$
- (Transitivity):
$$\frac{\Gamma \vdash t_1 \sim t_2 \quad \Gamma \vdash t_2 \sim t_3}{\Gamma \vdash t_1 \sim t_3}$$

系统满足假设规则：

- (Hypothesis): 若 $(t_1 \sim t_2) \in \Gamma$ ，则 $\Gamma \vdash t_1 \sim t_2$

此外，系统包含以下关于连接符 $:=$ 的规则：

- (Congruence):
$$\frac{\Gamma \vdash t_1 \sim s_1 \quad \Gamma \vdash t_2 \sim s_2}{\Gamma \vdash (t_1 := t_2) \sim (s_1 := s_2)}$$

- (Def-Elim): $\frac{\Gamma \vdash (t_1 := t_2) \sim ()}{\Gamma \vdash t_1 \sim t_2}$
- (Def-Reflection): $\frac{\Gamma \vdash t_1 \sim t_2}{\Gamma \vdash (t_1 := t_2) \sim ()}$
- (Reif): $\frac{}{\Gamma \vdash t \sim (t := ())}$

第 3 节 模型论语义 (Model-Theoretic Semantics) - \mathcal{M}_1

定义 3.1 (形式记号)

为确保语义定义的精确性，我们引入以下记号：

- $t_1 \equiv t_2$ 表示项 t_1 和 t_2 **语法同一 (syntactically identical)**。
- $v_1 = v_2$ 表示值 v_1 和 v_2 **语义相等 (semantically equal)**。由于最终求得的值是规范形式的项，此处的相等即为语法同一。

定义 3.2 (世界描述与模型构造)

- **指称断言 (Referential Assertion)** 是一个形如 $(\text{ref } s \ t)$ 的有序对，其中 $s \in \mathcal{S}$ 且 $t \in \mathcal{T}$ 。
- **世界描述 \mathcal{W}** 是一个由指称断言组成的有限集合。
- **初始环境 $\mathcal{E}_{\mathcal{W}}: \mathcal{S} \rightarrow \mathcal{T}$** 是由 \mathcal{W} 定义的偏函数： $\mathcal{E}_{\mathcal{W}}(s) = t \quad (\text{ref } s \ t) \in \mathcal{W}$ 。
- **\mathcal{M}_1 模型 $\mathcal{M}_{1,\mathcal{W}}$** 是一个全函数 $\mathcal{M}: \mathcal{S} \rightarrow \mathcal{T}$ ，它通过解析环境 $\mathcal{E}_{\mathcal{W}}$ 中的指称链来确定每个符号的最终指称对象。该函数构造如下：

对任意符号 $s_0 \in \mathcal{S}$ ，我们定义其在模型 \mathcal{M} 中的值 $\mathcal{M}(s_0)$ 如下：

1. 构造一个序列 $\langle s_0, s_1, s_2, \dots \rangle$ ，其中 $s_1 = \mathcal{E}_{\mathcal{W}}(s_0)$ ，只要 $s \in \mathcal{S}$ 且 $\mathcal{E}_{\mathcal{W}}(s)$ 有定义。
2. 由于 \mathcal{W} 是有限的， $\mathcal{E}_{\mathcal{W}}$ 的定义域 $\text{dom}(\mathcal{E}_{\mathcal{W}})$ 也是一个有限集。因此，序列中的符号必然来自 $\text{dom}(\mathcal{E}_{\mathcal{W}}) \cup \{s_0\}$ 。根据鸽巢原理，该序列必然在有限步内终止于一个不在 $\text{dom}(\mathcal{E}_{\mathcal{W}})$ 中的项，或在 $\text{dom}(\mathcal{E}_{\mathcal{W}})$ 内进入循环。

终止情况: 序列 P 形如 s_0, \dots, s, t ，其中 $t \notin \mathcal{S}$ 或 $t \notin \text{dom}(\mathcal{E}_{\mathcal{W}})$ 。此时，我们定义 $\mathcal{M}(s_0) = t$ 。

循环情况: 序列 P 形如 s_0, \dots, s, \dots, s ，其中 $s = s_0$ 且 $s \in \text{dom}(\mathcal{E}_{\mathcal{W}})$ 。这表示检测到一个直接的指称循环。此时，我们定义 $\mathcal{M}(s_0) = \perp$ 。

该构造过程对每个 $s_0 \in \mathcal{S}$ 都是确定且终止的，因此模型 $\mathcal{M}_{1,\mathcal{W}}$ 是良定义的。为简洁，下文将 $\mathcal{M}_{1,\mathcal{W}}$ 简记为 \mathcal{M} 。

定义 3.3 (语义求值)

项的语义求值分为两步：规范化和求值。

1. **规范形式的集合 (Set of Canonical Forms, or Values):** 一个项的中间规范形式 (或称值)，记为 v ，属于集合 \mathcal{V} 。该集合是满足以下条件的最小集合：
 - 若 $s \in \mathcal{S}$ ，则 $s \in \mathcal{V}$ 。
 - 原子常量 $()$ 、 \perp 和 \perp_cycle 均属于 \mathcal{V} 。
 - 若 $v_1, v_2 \in \mathcal{V}$ ，则 $(v_1 := v_2) \in \mathcal{V}$ 。
 注意，此定义在语法上与项集 \mathcal{T} 等价，即 $\mathcal{V} = \mathcal{T}$ 。在此语境下，集合 \mathcal{V} 特指作为规范

化 (NF) 过程的输出域和求值 (Eval) 过程的输入域的项集。

2. **规范化 (Normalization)**: 函数 $NF(t, M)$ 将一个项 t 转化为其规范形式, 该形式是一个属于 \mathcal{V} 的值。此过程通过对 t 的结构进行递归, 将项中出现的所有符号 s 替换为它们在模型 M 中的最终指称对象, 并递归地规范化结果。为处理由项结构 (如 $s := (t := s)$) 引入的间接循环, 规范化过程需记录当前正在展开的符号。

我们定义主函数 $NF(t, M)$ 如下:

$$N(t, \mathcal{M}) = N(t, \mathcal{M}, \emptyset)$$

其中辅助函数 $NF'(t, M, V)$ (V 是已访问符号集) 定义如下:

$$N(t, \mathcal{M}, V) = \perp \quad \text{若 } t \in \mathcal{S} \text{ 且 } t \in N(\mathcal{M}(t), \mathcal{M}, \{t\}) \quad \text{若 } t \in \mathcal{S} \text{ 且 } t \neq \perp \quad \text{若 } t \text{ 是原子常量 } (), \perp, \text{ 或 } \perp_e$$

此定义确保了规范化过程对于任何输入项 t 和模型 M 总是终止的。 $NF(t, M)$ 的结果是一个属于 \mathcal{V} 的值。

3. **求值 (Evaluation)**: 一个项 $t \in \mathcal{T}$ 的最终值 $\llbracket t \rrbracket_M \in \mathcal{V}$ 定义如下:

令 $v_t = N(t, \mathcal{M})$ 。

$$\llbracket t \rrbracket_M = a(v_t)$$

其中 $Eval$ 函数定义在值 $v \in \mathcal{V}$ 上:

$$a(v) = v \quad \text{若 } v \in \mathcal{S} \text{ 或 } v \text{ 是原子常量 } (), \perp, \perp_e() \quad \text{若 } v = (v_1 := v_2) \text{ 且 } a(v_1) = a(v_2)a(v_1) \quad \text{若 } v = (v_1$$

在第四种情况中, 结果 $(a(v_1) := a(v_2))$ 被称为**滞留项 (stuck term)**。注意, $Eval$ 函数的递归调用作用于项的子结构, 因此它总是终止的。最终求得的值是 $Eval$ 函数的一个不动点。

定义 3.4 (满足与蕴含)

- **模型满足**: $\mathcal{M} \models t_1 \sim t_2 \iff \llbracket t_1 \rrbracket_M = \llbracket t_2 \rrbracket_M$ 。
- **语义蕴含**: $\Gamma \models t_1 \sim t_2 \iff \mathcal{M} \models t_1 \sim t_2$ (若 $\mathcal{M} \models \Gamma$, 则 $\mathcal{M} \models t_1 \sim t_2$)。

第 4 节 连接符 $:=$ 的性质

性质 4.1 (非交换性)

连接符 $:=$ 不满足交换律。即, 一般情况下 $\llbracket (t_1 := t_2) \rrbracket_M \neq \llbracket (t_2 := t_1) \rrbracket_M$ 。

- **证明:** 考虑一个空世界描述 $\mathcal{W} =$ 的模型 \mathcal{M} ，以及两个不同的符号 $s_1, s_2 \in \mathcal{S}$ ，其中 $s_1 \neq s_2$ 。在此模型中，对于任何符号 s ， $\mathcal{M}(s) = s$ 。

1. 求值 $\llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}}$:

- 规范化: $N((s_1 := s_2), \mathcal{M}) = (N(s_1, \mathcal{M}) := N(s_2, \mathcal{M})) = (s_1 := s_2)$ 。
- 令 $v = (s_1 := s_2)$ 。我们计算 $a(v)$ 。
- 首先，计算子项的值: $a(s_1) = s_1$ 且 $a(s_2) = s_2$ 。
- 由于 $s_1 \neq s_2$ ，我们有 $a(s_1) \neq a(s_2)$ 。
- 同时， $a(s_2) \neq ()$ 。
- 因此，这属于求值规则的第四种情况（滞留项）。
- $\llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}} = (a(s_1) := a(s_2)) = (s_1 := s_2)$ 。

2. 求值 $\llbracket (s_2 := s_1) \rrbracket_{\mathcal{M}}$:

- 同理，规范化得到 $(s_2 := s_1)$ 。
- 求值得 $\llbracket (s_2 := s_1) \rrbracket_{\mathcal{M}} = (s_2 := s_1)$ 。

3. 由于 $(s_1 := s_2) \neq (s_2 := s_1)$ ，它们的值不相等。证毕。

性质 4.2 (非结合性)

连接符 $:=$ 不满足结合律。即，一般情况下 $\llbracket ((t_1 := t_2) := t_3) \rrbracket_{\mathcal{M}} \neq \llbracket (t_1 := (t_2 := t_3)) \rrbracket_{\mathcal{M}}$ 。

- **证明:** 考虑与上述相同的模型 \mathcal{M} 和三个不同的符号 s_1, s_2, s_3 。

1. LHS: $\llbracket ((s_1 := s_2) := s_3) \rrbracket_{\mathcal{M}}$

- 内部项的值 $\llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}} = (s_1 := s_2)$ 。
- 外部项求值为 $\llbracket ((s_1 := s_2) := s_3) \rrbracket_{\mathcal{M}}$ 。其规范形式为 $((s_1 := s_2) := s_3)$ 。由于 $\llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}} \neq s_3$ 且 $\llbracket s_3 \rrbracket_{\mathcal{M}} \neq ()$ ，结果为滞留项 $((s_1 := s_2) := s_3)$ 。

2. RHS: $\llbracket (s_1 := (s_2 := s_3)) \rrbracket_{\mathcal{M}}$

- 内部项的值 $\llbracket (s_2 := s_3) \rrbracket_{\mathcal{M}} = (s_2 := s_3)$ 。
- 外部项求值为 $\llbracket (s_1 := (s_2 := s_3)) \rrbracket_{\mathcal{M}}$ 。同理，结果为滞留项 $(s_1 := (s_2 := s_3))$ 。

3. LHS 和 RHS 的结果是两个语法上不同的嵌套项，因此它们的值不相等。证毕。

第 5 节 系统可靠性 (Soundness)

定理 5.1 (可靠性)

本系统的证明论是可靠的。即，若 $\Gamma \vdash t_1 \sim t_2$ ，则 $\Gamma \models t_1 \sim t_2$ 。

- **证明思路:** 我们通过归纳证明每一条推理规则都是保真 (truth-preserving) 的。对于公理（无前提的规则），我们证明其结论在任何模型下都为真。对于有前提的规则，我们假设其前提在语义上成立，并证明其结论也必然成立。语义等价关系 \sim 显然满足反身性、对称性和传递性，因此对应的证明论规则是可靠的。我们重点关注其余四条规则。

引理 5.1.1 (Reif 可靠性)

规则 (Reif) 是可靠的。

- **待证:** $t \sim (t := ())$ 对于任意 t 成立。

- **证明:** 取任意模型 \mathcal{M} 。需证 $\llbracket t \rrbracket_{\mathcal{M}} = \llbracket (t := ()) \rrbracket_{\mathcal{M}}$ 。

令 $v = \llbracket t \rrbracket_{\mathcal{M}}$ 。我们来计算 $\llbracket (t := ()) \rrbracket_{\mathcal{M}}$ 。

1. 规范化: $N((t := ()), \mathcal{M}) \ (N(t, \mathcal{M}) := N((), \mathcal{M})) \ (N(t, \mathcal{M}) := ())$ 。
2. 求值: 我们对 $a(N(t, \mathcal{M}) := ())$ 进行分情况讨论。令 $v_1 = a(N(t, \mathcal{M})) = \llbracket t \rrbracket_{\mathcal{M}}$ 且 $v_2 = a(()) = ()$ 。

情况 A: $v_1 = v_2$, 即 $\llbracket t \rrbracket_{\mathcal{M}} = ()$ 。根据定义 3.3 第一条, $\llbracket (t := ()) \rrbracket_{\mathcal{M}} = ()$ 。此时左右两边相等。

情况 B: $v_1 \neq v_2$, 即 $\llbracket t \rrbracket_{\mathcal{M}} \neq ()$ 。根据定义 3.3 第二条, $\llbracket (t := ()) \rrbracket_{\mathcal{M}} = v_1 = \llbracket t \rrbracket_{\mathcal{M}}$ 。此时左右两边也相等。

由于所有情况均成立, 该规则可靠。证毕。

引理 5.1.2 (Def-Reflection 可靠性)

规则 (Def-Reflection) 是可靠的。

- **待证:** 若 $\Gamma \ t_1 \sim t_2$, 则 $\Gamma \ (t_1 := t_2) \sim ()$ 。
- **证明:** 假设 $\Gamma \ t_1 \sim t_2$ 。取任意满足 $\mathcal{M} \models \Gamma$ 的模型 \mathcal{M} 。根据假设, 必有 $\mathcal{M} \ t_1 \sim t_2$, 即 $\llbracket t_1 \rrbracket_{\mathcal{M}} = \llbracket t_2 \rrbracket_{\mathcal{M}}$ 。根据定义 3.3 中求值规则的第二条, 此条件直接导出 $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = ()$ 。又因 $\llbracket () \rrbracket_{\mathcal{M}} = ()$, 故 $\mathcal{M} \ (t_1 := t_2) \sim ()$ 。证毕。

引理 5.1.3 (Def-Elim 可靠性)

规则 (Def-Elim) 是可靠的。

- **待证:** 若 $\Gamma \ (t_1 := t_2) \sim ()$, 则 $\Gamma \ t_1 \sim t_2$ 。
- **证明:** 假设 $\Gamma \ (t_1 := t_2) \sim ()$ 。取任意满足 $\mathcal{M} \models \Gamma$ 的模型 \mathcal{M} 。根据假设, 必有 $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = \llbracket () \rrbracket_{\mathcal{M}} = ()$ 。考察定义 3.3 的求值规则, 使得 $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}}$ 值为 $()$ 的唯一条件是 $\llbracket t_1 \rrbracket_{\mathcal{M}} = \llbracket t_2 \rrbracket_{\mathcal{M}}$ 。因此, $\mathcal{M} \ t_1 \sim t_2$ 。证毕。

引理 5.1.4 (Congruence 可靠性)

规则 (Congruence) 是可靠的。

- **待证:** 若 $\Gamma \models t_1 \sim s_1$ 且 $\Gamma \models t_2 \sim s_2$, 则 $\Gamma \models (t_1 := t_2) \sim (s_1 := s_2)$ 。
- **证明:** 假设前提成立。取任意满足 $\mathcal{M} \models \Gamma$ 的模型 \mathcal{M} 。根据假设, 我们有:

1. $\llbracket t_1 \rrbracket_{\mathcal{M}} = \llbracket s_1 \rrbracket_{\mathcal{M}}$
2. $\llbracket t_2 \rrbracket_{\mathcal{M}} = \llbracket s_2 \rrbracket_{\mathcal{M}}$

我们的目标是证明 $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = \llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}}$ 。

令 $v_1 = \llbracket t_1 \rrbracket_{\mathcal{M}}$, $v_2 = \llbracket t_2 \rrbracket_{\mathcal{M}}$, $v_1 = \llbracket s_1 \rrbracket_{\mathcal{M}}$, $v_2 = \llbracket s_2 \rrbracket_{\mathcal{M}}$ 。前提即 $v_1 = v_1$ 且 $v_2 = v_2$ 。

根据定义 3.3 的四分支求值规则, 对 $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}}$ 的求值结果进行分情况讨论。这是一个对所有可能值的完全划分。

- **情况 A:** $v_1 = v_2$ 。
 - 根据前提, 通过传递性有 $v_1 = v_1 = v_2 = v_2$, 因此 $v_1 = v_2$ 。
 - 根据求值规则第二条:
 - $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = ()$

- $\llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}} = ()$
- 两者相等，结论成立。
- **情况 B:** $v_1 \ v_2$ 且 $v_2 = ()$ 。
 - 由于前提保证 $v_1 = v_1$ 且 $v_2 = v_2$ ，任何关于 v_1, v_2 的等式或不等式判断都直接传递给 v_1, v_2 。因此，我们有 $v_1 \ v_2$ 且 $v_2 = ()$ 。这意味着对 $(s_1 := s_2)$ 的求值也遵循此情况的规则。
 - 根据求值规则第三条：
 - $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = v_1$
 - $\llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}} = v_1$
 - 由于前提保证了 $v_1 = v_1$ ，因此两者相等，结论成立。
- **情况 C:** 其他所有情况 (即 $v_1 \ v_2$ 且 $v_2 \ ()$)。
 - 同样，根据前提可推断出 $v_1 \ v_2$ 且 $v_2 \ ()$ 。
 - 根据求值规则第四条，结果为滞留项：
 - $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = (v_1 := v_2)$
 - $\llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}} = (v_1 := v_2)$
 - 我们已知前提为 $v_1 = v_1$ 和 $v_2 = v_2$ 。由于值本身是规范形式的项，这意味着 $v_1 \ v_1$ 且 $v_2 \ v_2$ 。
 - 将此代入，我们发现构造出的两个滞留项 $(v_1 := v_2)$ 和 $(v_1 := v_2)$ 在语法上是完全相同的。
 - 因此，两者相等，结论成立。

由于在所有可能的情况下 $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = \llbracket (s_1 := s_2) \rrbracket_{\mathcal{M}}$ 都成立，Congruence 规则是可靠的。证毕。

引理 5.1.5 (Hypothesis 可靠性)

规则 (Hypothesis) 是可靠的。

- **证明:** 我们需要证明，若一个判断 $\Gamma \vdash t_1 \sim t_2$ 是通过 (Hypothesis) 规则得出的，则 $\Gamma \models t_1 \sim t_2$ 成立。
 1. 根据 (Hypothesis) 规则的定义， $\Gamma \vdash t_1 \sim t_2$ 成立的前提是 $(t_1 \sim t_2) \in \Gamma$ 。
 2. 根据语义蕴含的定义 (定义 3.4)， $\Gamma \models t_1 \sim t_2$ 意为：对于任何满足 $\mathcal{M} \models \Gamma$ 的模型 \mathcal{M} ，都有 $\mathcal{M} \models t_1 \sim t_2$ 。
 3. 取一个任意的模型 \mathcal{M} ，并假设 $\mathcal{M} \models \Gamma$ 。根据模型满足上下文的定义，这意味着对于 Γ 中的**每一个**判断 $(t_a \sim t_b)$ ，都有 $\mathcal{M} \models t_a \sim t_b$ 。
 4. 由于 $(t_1 \sim t_2) \in \Gamma$ ，因此模型 \mathcal{M} 必然满足该判断，即 $\mathcal{M} \models t_1 \sim t_2$ 。
 5. 因为 \mathcal{M} 是任意选择的，所以此结论对所有满足 Γ 的模型都成立。这正是 $\Gamma \models t_1 \sim t_2$ 的语义。

因此，(Hypothesis) 规则是可靠的。证毕。

定理 5.1 (可靠性)

本系统的证明论是可靠的。即，若 $\Gamma \vdash t_1 \sim t_2$ ，则 $\Gamma \models t_1 \sim t_2$ 。

- **证明:** 我们通过对 $\Gamma \vdash t_1 \sim t_2$ 的推导结构进行归纳来证明此定理。这意味着我们需要证明系统中所有的公理（无前提的规则）都是有效的，并且所有推理规则都是保真（truth-preserving）的。

- **基本情况 (公理):**

- 对于 (Reflexivity) 和 (Reif)，其结论的普遍有效性已在引理 5.1.1 的证明思路中得到论证。
- 对于 (Hypothesis)，其可靠性已在**引理 5.1.5** 中证明。

- **归纳步骤 (推理规则):**

- 对于 (Symmetry)、(Transitivity)、(Congruence)、(Def-Elim) 和 (Def-Reflection)，我们需要假设其前提在语义上成立，并证明其结论也必然成立。这些规则的可靠性已分别在**引理 5.1.1 至 5.1.4** 的证明中得到验证。

由于推导的每个可能步骤（无论是应用公理还是推理规则）都保持了语义的有效性，因此整个推导过程是可靠的。若 $\Gamma \vdash t_1 \sim t_2$ 可被推导，则 $\Gamma \models t_1 \sim t_2$ 必然成立。**证毕。**

第 7 节 系统一致性 (Consistency)

定理 7.1 (一致性)

LRS 系统是一致的。

- **证明:** 一个形式系统是一致的，当且仅当并非所有合法的判断都是可证的。我们将通过证明存在一个不可证的判断 $\vdash \perp \sim ()$ 来证明系统的一致性。（此处 \vdash 表示从空上下文 $\Gamma = \emptyset$ 进行推导）。
 1. 根据**可靠性定理 (定理 5.1)**，若 $\Gamma \vdash t_1 \sim t_2$ ，则 $\Gamma \models t_1 \sim t_2$ 。
 2. 其逆否命题为：若 $\Gamma \not\models t_1 \sim t_2$ ，则 $\Gamma \not\vdash t_1 \sim t_2$ 。
 3. 我们构造一个具体的模型 M_\emptyset ，该模型基于一个空的世界描述 $W = \emptyset$ 。
 4. 在该模型中，我们计算 \perp 和 $()$ 的值：
 - $[\perp]_{M_\emptyset} = \text{Eval}(\text{NF}(\perp, M_\emptyset)) = \text{Eval}(\perp) = \perp$ 。
 - $[()]_{M_\emptyset} = \text{Eval}(\text{NF}(), M_\emptyset) = \text{Eval}(()) = ()$ 。
 5. 由于 \perp 和 $()$ 是两个语法上不同的项，我们有 $[\perp]_{M_\emptyset} \neq [()]_{M_\emptyset}$ 。
 6. 根据**模型满足 (定义 3.4)**，这意味着 $M_\emptyset \not\models \perp \sim ()$ 。
 7. 由于存在至少一个模型不满足该判断，因此该判断不是普遍有效的，即 $\not\models \perp \sim ()$ 。对于空上下文 $\Gamma = \emptyset$ ，这等价于 $\emptyset \not\models \perp \sim ()$ 。
 8. 根据可靠性定理的逆否命题，我们得出结论： $\emptyset \not\vdash \perp \sim ()$ 。
 9. 需要注意的是，新加入的 (Hypothesis) 规则仅在上下文 Γ 非空时才可应用。在我们考虑的 $\Gamma = \emptyset$ 的情况下，此规则无法被调用。因此，该规则不影响从空上下文中可证明的判断集合。
 10. 因为存在一个不可被证明的判断，所以 LRS 系统是一致的。**证毕。**

第 8 节 动机与核心元定理

定理 8.1 (定义性完备性元定理)

对于任意一个模型 \mathcal{M} 和任意两个项 $t_1, t_2 \in \mathcal{T}$ ，以下两个陈述是等价的：

1. 语义等价 (Semantic Equivalence): $\mathcal{M} \models t_1 \sim t_2$
2. 定义性成功 (Definitional Success): $\mathcal{M} \models (t_1 := t_2) \sim ()$

证明:

该定理是模型论语义定义（特别是定义 3.3 和 3.4）的直接推论。

- (1 \Rightarrow 2): 假设 $\mathcal{M} \models t_1 \sim t_2$ 。根据定义 3.4，这意味着 $\llbracket t_1 \rrbracket_{\mathcal{M}} = \llbracket t_2 \rrbracket_{\mathcal{M}}$ 。根据定义 3.3 中对定义项求值的第二条规则，此条件直接导出 $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = ()$ 。由于 $\llbracket () \rrbracket_{\mathcal{M}}$ 也为 $()$ ，因此 $\mathcal{M} \models (t_1 := t_2) \sim ()$ 。
- (2 \Rightarrow 1): 假设 $\mathcal{M} \models (t_1 := t_2) \sim ()$ 。根据定义 3.4，这意味着 $\llbracket (t_1 := t_2) \rrbracket_{\mathcal{M}} = \llbracket () \rrbracket_{\mathcal{M}} = ()$ 。考察定义 3.3 中对定义项求值的所有情况，能够使结果为 $()$ 的唯一条件是 $\llbracket t_1 \rrbracket_{\mathcal{M}} = \llbracket t_2 \rrbracket_{\mathcal{M}}$ 。这正是 $\mathcal{M} \models t_1 \sim t_2$ 的定义。

因此，两个陈述等价。证毕。

第 9 节 句法求值

引理 9.1.1 (值的句法求值)

对于任意值 $v \in V$ ， $\vdash v \sim \text{Eval}(v)$ 。

- 证明: 我们对值 v 的结构进行归纳。
 - 基本情况: v 是一个原子常量 $()$ 、 \perp 或 \perp_cycle ，或一个符号 $s \in S$ 。根据 Eval 的定义， $\text{Eval}(v) = v$ 。根据公理 (Reflexivity)， $\vdash v \sim v$ 成立。因此 $\vdash v \sim \text{Eval}(v)$ 。
 - 归纳步骤: 假设对于所有比 v 结构更简单的值 v' ， $\vdash v' \sim \text{Eval}(v')$ 均成立（归纳假设 IH）。考虑 v 的形式为 $(v_1 := v_2)$ 。根据归纳假设，我们有：
 - (1) $\vdash v_1 \sim \text{Eval}(v_1)$
 - (2) $\vdash v_2 \sim \text{Eval}(v_2)$

我们根据 $\text{Eval}(v)$ 的定义进行分情况讨论：

- 情况 A: $\text{Eval}(v_1) = \text{Eval}(v_2)$ 。此时 $\text{Eval}(v) = ()$ 。
 1. 从 (1) 和 (2)，通过 Symmetry 和 Transitivity，并利用 $\text{Eval}(v_1) = \text{Eval}(v_2)$ ，可推导出 $\vdash v_1 \sim v_2$ 。
 2. 对 $\vdash v_1 \sim v_2$ 应用 (Def-Reflection)，得到 $\vdash (v_1 := v_2) \sim ()$ 。
 3. 此即 $\vdash v \sim \text{Eval}(v)$ 。
- 情况 B: $\text{Eval}(v_1) \neq \text{Eval}(v_2)$ 且 $\text{Eval}(v_2) = ()$ 。此时 $\text{Eval}(v) = \text{Eval}(v_1)$ 。
 1. 由 (2) 和 $\text{Eval}(v_2) = ()$ 可知 $\vdash v_2 \sim ()$ 。

2. 对 $(1) \vdash v_1 \sim \text{Eval}(v_1)$ 和 $\vdash v_2 \sim ()$ 应用 (Congruence)，得到 $\vdash (v_1 := v_2) \sim (\text{Eval}(v_1) := ())$ 。
 3. 根据公理 (Reif)，我们有 $\vdash \text{Eval}(v_1) \sim (\text{Eval}(v_1) := ())$ 。
 4. 应用 (Symmetry)，得到 $\vdash (\text{Eval}(v_1) := ()) \sim \text{Eval}(v_1)$ 。
 5. 将第 2 步和第 4 步的结果通过 (Transitivity) 结合，得到 $\vdash (v_1 := v_2) \sim \text{Eval}(v_1)$ 。
 6. 此即 $\vdash v \sim \text{Eval}(v)$ 。
- **情况 C: 其他所有情况。** 此时 $\text{Eval}(v) = (\text{Eval}(v_1) := \text{Eval}(v_2))$ 。
 1. 直接对 $(1) \vdash v_1 \sim \text{Eval}(v_1)$ 和 $(2) \vdash v_2 \sim \text{Eval}(v_2)$ 应用 (Congruence) 规则。
 2. 我们立即得到 $\vdash (v_1 := v_2) \sim (\text{Eval}(v_1) := \text{Eval}(v_2))$ 。
 3. 此即 $\vdash v \sim \text{Eval}(v)$ 。

由于所有情况均已覆盖，引理得证。**证毕。**

引理 9.1.2 (Eval 的幂等性)

对于任意值 $v \in V$ ， $\text{Eval}(\text{Eval}(v)) = \text{Eval}(v)$ 。

- **证明:** 令 $v' = \text{Eval}(v)$ 。根据 Eval 的定义， v' 要么是一个符号/原子常量，要么是一个“滞留项” $(v_1' := v_2')$ ，其中 v_1' 和 v_2' 本身也是 Eval 的输出。在所有这些情况下， v' 都是一个**范式值** (canonical value)，即 Eval 的不动点。对其再次应用 Eval 不会改变其值，故 $\text{Eval}(v') = v'$ 。因此， $\text{Eval}(\text{Eval}(v)) = \text{Eval}(v)$ 。**证毕。**

第 10 节 完备性 (Completeness)

本节致力于证明 LRS 系统的证明论对于其模型论语义是强完备的。我们将采用标准的 Henkin 风格证明，其核心在于构造一个“典范模型”，该模型由一个极大一致的判断集合（上下文）生成。

10.1 极大一致集与 Henkin 集

定义 10.1 (一致性与极大一致性)

1. 一个上下文 Γ 是**一致的 (consistent)**，当且仅当 $\Gamma \not\vdash \perp \sim ()$ 。
2. 一个上下文 Γ^* 是**极大一致的 (maximally consistent)**，如果 Γ^* 是一致的，并且对于任何不属于 Γ^* 的判断 $\phi \notin \Gamma^*$ ，集合 $\Gamma^* \cup \{\phi\}$ 都是不一致的。

引理 10.2 (极大一致集的性质)

若 Γ^* 是一个极大一致集，则它具有以下性质：

1. **演绎闭包:** 对于任意判断 ϕ ， $\Gamma^* \vdash \phi$ 当且仅当 $\phi \in \Gamma^*$ 。
2. **否定完备性:** 对于任意判断 ϕ ，若 $\phi \notin \Gamma^*$ ，则 $\Gamma^* \cup \{\phi\}$ 不一致，即 $\Gamma^*, \phi \vdash \perp \sim ()$ 。

3. **等价关系闭包**: Γ^* 中由 \sim 诱导的关系是等价关系。例如, 若 $(t_1 \sim t_2) \in \Gamma^*$ 且 $(t_2 \sim t_3) \in \Gamma^*$, 则 $(t_1 \sim t_3) \in \Gamma^*$ 。

证明: 这些是极大一致集在任何具备基本演绎规则的逻辑系统中的标准性质。

定义 10.3 (Henkin 集)

一个上下文 Γ^* 是一个 **Henkin 集**, 如果它同时是一个极大一致集, 并且满足**典范值属性 (Canonical Value Property)**: 对任意项 $t \in T$, 存在一个值 $v \in V$ 使得判断 $(t \sim v)$ 属于 Γ^* 。

10.2 完备性证明的核心引理

引理 10.4 (可分辨范式值)

令 v_1, v_2 为两个**范式值 (canonical values)**, 即 $v_1, v_2 \in V$ 且 $\text{Eval}(v_1) \equiv v_1$ 且 $\text{Eval}(v_2) \equiv v_2$ 。如果 v_1 和 v_2 在句法上不同 ($v_1 <\text{binary data, 2 bytes}> v_2$), 那么 $\not\models v_1 \sim v_2$ 。

• **证明:**

1. 根据**可靠性定理 (5.1)**, 若 $\vdash v_1 \sim v_2$, 则 $\models v_1 \sim v_2$ 。我们将证明其逆否命题。
 2. 我们需证明 $\not\models v_1 \sim v_2$ 。为此, 我们构造一个反例模型。考虑基于空世界描述 $W = \emptyset$ 的模型 M_\emptyset 。
 3. 对于任何值 $v \in V$, 其规范形式 $\text{NF}(v, M_\emptyset) \equiv v$ 。
 4. 因此, $[v]_{M_\emptyset} = \text{Eval}(\text{NF}(v, M_\emptyset)) = \text{Eval}(v)$ 。
 5. 根据引理假设, v_1 和 v_2 是范式值, 故 $[v_1]_{M_\emptyset} = v_1$ 且 $[v_2]_{M_\emptyset} = v_2$ 。
 6. 又因假设 $v_1 <\text{binary data, 2 bytes}> v_2$, 我们有 $[v_1]_{M_\emptyset} \neq [v_2]_{M_\emptyset}$ 。
 7. 根据模型满足的定义, 这意味着 $M_\emptyset \not\models v_1 \sim v_2$, 因此 $\not\models v_1 \sim v_2$ 。
 8. 由可靠性定理可知, $\not\models v_1 \sim v_2$ 。
- 证毕。**

引理 10.5 (典范值见证引理)

令 Γ' 为一个一致的上下文。对于任意项 $t \in T$, 存在一个值 $v \in V$ 使得上下文 $\Gamma' \cup \{(t \sim v)\}$ 是一致的。

• **证明:**

我们通过归谬法证明。假设引理为假, 则存在一个非空的**反例项集** T_{fail} , 其成员 t 满足: 存在一个一致的上下文 Γ' 使得对所有 $v \in V$, $\Gamma' \cup \{(t \sim v)\}$ 均不一致。

由于 T_{fail} 非空, 根据良序原则 (基于项的结构复杂度), 存在一个复杂度最小的项 $t_0 \in T_{\text{fail}}$ 。根据 t_0 的定义, 存在一个一致的上下文 Γ_0 。使得:

(*) 对所有 $v \in V$, $\Gamma_0 \cup \{(t_0 \sim v)\}$ 是不一致的 (即 $\Gamma_0, (t_0 \sim v) \vdash \perp \sim ()$)。

我们对 t_0 的结构进行分情况讨论:

• **情况 1: t_0 是一个值 ($t_0 \in V$)**

1. 令 $t_0 \equiv v'$ 。根据**引理 9.1.1**, $\vdash v' \sim \text{Eval}(v')$ 。

2. 令 $v_eval = Eval(v')$ 。 v_eval 是一个值。
3. 将 $v = v_eval$ 代入假设 (*), 可知 $\Gamma_0 \cup \{(v' \sim v_eval)\}$ 不一致。
4. 这意味着 $\Gamma_0, (v' \sim v_eval) \vdash \perp \sim ()$ 。
5. 但我们有 $\Gamma_0 \vdash v' \sim v_eval$ (因为它是公理的推论)。
6. 根据演绎系统的切除性质, 若 $\Gamma_0 \vdash \phi$ 且 $\Gamma_0, \phi \vdash \psi$, 则 $\Gamma_0 \vdash \psi$ 。
7. 因此, $\Gamma_0 \vdash \perp \sim ()$, 表明 Γ_0 本身不一致。这与 Γ_0 的前提相矛盾。故 t_0 不可能是一个值。

• **情况 2: t_0 是一个复合项 ($t_0 \equiv (t_1 := t_2)$)**

1. 由于 t_1 和 t_2 的复杂度均小于 t_0 , 它们不属于 T_fail 。
2. 因为 Γ_0 一致且 $t_1 \notin T_fail$, 存在值 $v_1 \in V$ 使得 $\Gamma_1 = \Gamma_0 \cup \{(t_1 \sim v_1)\}$ 是一致的。
3. 因为 Γ_1 一致且 $t_2 \notin T_fail$, 存在值 $v_2 \in V$ 使得 $\Gamma_2 = \Gamma_1 \cup \{(t_2 \sim v_2)\}$ 是一致的。
4. 在一致的上下文 Γ_2 中, 我们有 $\Gamma_2 \vdash t_1 \sim v_1$ 和 $\Gamma_2 \vdash t_2 \sim v_2$ 。
5. 通过 (Congruence) 规则, $\Gamma_2 \vdash (t_1 := t_2) \sim (v_1 := v_2)$, 即 $\Gamma_2 \vdash t_0 \sim (v_1 := v_2)$ 。
6. 令 $v_final = Eval((v_1 := v_2))$ 。 v_final 是一个值。根据引理 9.1.1 和传递性, $\Gamma_2 \vdash t_0 \sim v_final$ 。
7. 若 $\Gamma_2 \cup \{(t_0 \sim v_final)\}$ 不一致, 则 Γ_2 自身必不一致。但我们已证明 Γ_2 一致。
8. 因此, $\Gamma_2 \cup \{(t_0 \sim v_final)\}$ 是一致的。
9. 该集合包含 $\Gamma_0 \cup \{(t_0 \sim v_final)\}$ 作为子集, 因此 $\Gamma_0 \cup \{(t_0 \sim v_final)\}$ 也必须是一致的。
10. 这与 t_0 的定义 (*) 相矛盾。故 t_0 不可能是一个复合项。

• **情况 3: t_0 是一个符号 ($t_0 \equiv s$)**

1. 若 T_fail 非空, 其最小复杂度元素 t_0 必然是符号 s 。
2. 因此, 存在一致的 Γ_0 使得 $\forall v \in V, \Gamma_0 \cup \{(s \sim v)\}$ 不一致。
3. 考虑项 $t' = (s := ())$ 。根据 (Reif) 规则, $\vdash s \sim t'$, 因此 $\Gamma_0 \vdash s \sim t'$ 。
4. 我们证明 t' 也在 T_fail 中。为此, 我们必须证明对于所有 $v \in V$, $\Gamma_0 \cup \{(t' \sim v)\}$ 是不一致的。
5. 我们通过反证法证明。假设存在某个 $v \in V$ 使得 $\Gamma' = \Gamma_0 \cup \{(t' \sim v)\}$ 是一致的。
6. 根据 (Hypothesis) 规则, $\Gamma' \vdash t' \sim v$ 。
7. 由于 $\Gamma_0 \subset \Gamma'$, 我们也有 $\Gamma' \vdash s \sim t'$ 。
8. 通过 (Symmetry) 和 (Transitivity), 我们得到 $\Gamma' \vdash s \sim v$ 。
9. 根据初始假设 (*), 我们知道 $\Gamma_0, (s \sim v) \vdash \perp \sim ()$ 。
10. 由于 $\Gamma_0 \subset \Gamma'$ 且我们已推导出 $\Gamma' \vdash s \sim v$, 根据演绎系统的切除性质, 可推断出 $\Gamma' \vdash \perp \sim ()$ 。
11. 这意味着 Γ' 是不一致的, 与我们在步骤 5 的假设相矛盾。

12. 因此, 对所有 $v \in V$, $\Gamma \cup \{(t' \sim v)\}$ 都是不一致的。这表明 $t' \in T_{fail}$ 。

13. 然而, t' 的复杂度高于 s 。这与 s 是 T_{fail} 中复杂度最小的元素相矛盾。故 t 。不可能是一个符号。

由于 t 。既不是值, 也不是复合项, 也不是符号, 所以 t 。不存在。 T_{fail} 必须为空集。

证毕。

定理 10.6 (Henkin-Lindenbaum 引理)

任何一致的上下文 Γ 都可以被扩展为一个 Henkin 集 Γ^* , 使得 $\Gamma \subseteq \Gamma^*$ 。

• 证明:

1. 枚举: 枚举 L 。中所有的判断为 $\phi_0, \phi_1, \phi_2, \dots$, 所有的项为 $\tau_0, \tau_1, \tau_2, \dots$ 。

2. 构造: 递归定义上下文序列 $\Gamma_0, \Gamma_1, \Gamma_2, \dots$ 。

• $\Gamma_0 = \Gamma$ 。

• 奇数步骤 (极大性): $\Gamma_{2n+1} = \Gamma_{2n} \cup \{\phi_n\}$, 如果此集合一致; 否则 $\Gamma_{2n+1} = \Gamma_{2n}$ 。

• 偶数步骤 (Henkin 属性): 考虑项 τ_n 和一致的上下文 Γ_{2n+1} 。根据引理 10.5, 存在一个值 v 使得 $\Gamma_{2n+1} \cup \{(\tau_n \sim v)\}$ 是一致的。我们选择第一个这样的 v 并定义 $\Gamma_{2n+2} = \Gamma_{2n+1} \cup \{(\tau_n \sim v)\}$ 。

3. 最终集合: $\Gamma^* = \bigcup_{n \geq 0} \Gamma_n$ 。

4. 验证: Γ^* 包含 Γ , 并且通过构造, 它是一致的、极大一致的, 并且满足典范值属性。
证毕。

推论 10.7 (典范值的存在性与唯一性)

对于任意 Henkin 集 Γ^* 和任意项 $t \in T$, 存在一个唯一的**范式值** $v \in V$ (即 $\text{Eval}(v) \equiv v$), 使得 $(t \sim v) \in \Gamma^*$ 。

• 证明:

1. 存在性: 根据 Γ^* 的 Henkin 属性, 存在值 $v' \in V$ 使得 $(t \sim v') \in \Gamma^*$ 。令 $v = \text{Eval}(v')$ 。根据引理 9.1.1, $\vdash v' \sim v$, 由于 Γ^* 演绎闭包, $(v' \sim v) \in \Gamma^*$ 。通过传递性, $(t \sim v) \in \Gamma^*$ 。我们还需证明 v 是一个范式值。根据引理 9.1.2 (Eval 的幂等性), 我们有 $\text{Eval}(v) = \text{Eval}(\text{Eval}(v')) = \text{Eval}(v') = v$ 。因此 v 是一个范式值。

2. 唯一性: 假设存在两个不同的范式值 v_1, v_2 使得 $(t \sim v_1) \in \Gamma^*$ 和 $(t \sim v_2) \in \Gamma^*$ 。那么通过传递性, $(v_1 \sim v_2) \in \Gamma^*$ 。但根据引理 10.4, 由于 $v_1 <_{\text{binary data, 2 bytes}} v_2$, 我们有 $\not\vdash v_1 \sim v_2$ 。一个一致的集合不能证明一个不可证的判断, 因此 $(v_1 \sim v_2) \notin \Gamma^*$, 产生矛盾。故 $v_1 \equiv v_2$ 。

证毕。

我们将这个唯一的范式值记为 $\text{CV}_{\Gamma^*}(t)$ 。*

10.3 典范模型与强完备性证明

定义 10.8 (典范模型)

给定一个 Henkin 集 Γ^* ，我们构造**典范模型** M_{Γ^*} 如下：

对于每个符号 $s \in S$ ，我们定义 $M_{\Gamma^*}(s) = CV_{\{\Gamma^*\}}(s)$ 。

定理 10.9 (真理引理)

对于任意 Henkin 集 Γ^* 及其典范模型 M_{Γ^*} ，以及任意项 $t \in T$ ，以下等式成立：

$$\llbracket t \rrbracket = C(t)$$

- **证明:** 我们对项 t 的结构进行归纳。

- **基本情况 1:** t 是一个原子常量

$\llbracket t \rrbracket_{M_{\Gamma^*}} = \text{Eval}(t) = t$ 。同时 $CV_{\{\Gamma^*\}}(t) = t$ 。两者相等。

- **基本情况 2:** t 是一个符号 s

$\llbracket s \rrbracket_{M_{\Gamma^*}} = \text{Eval}(M_{\Gamma^*}(s))$ 。根据 M_{Γ^*} 的定义， $M_{\Gamma^*}(s) = CV_{\{\Gamma^*\}}(s)$ 。

因此 $\llbracket s \rrbracket_{M_{\Gamma^*}} = \text{Eval}(CV_{\{\Gamma^*\}}(s))$ 。由于 $CV_{\{\Gamma^*\}}(s)$ 是范式值，

$\text{Eval}(CV_{\{\Gamma^*\}}(s)) = CV_{\{\Gamma^*\}}(s)$ 。所以 $\llbracket s \rrbracket_{M_{\Gamma^*}} = CV_{\{\Gamma^*\}}(s)$ 。

- **归纳步骤:** t 的形式是 $(t_1 := t_2)$

归纳假设 (IH): $\llbracket t_1 \rrbracket_{M_{\Gamma^*}} = CV_{\{\Gamma^*\}}(t_1)$ 且 $\llbracket t_2 \rrbracket_{M_{\Gamma^*}} = CV_{\{\Gamma^*\}}(t_2)$ 。

语义求值: $\llbracket (t_1 := t_2) \rrbracket_{M_{\Gamma^*}} = \text{Eval}((\llbracket t_1 \rrbracket_{M_{\Gamma^*}} := \llbracket t_2 \rrbracket_{M_{\Gamma^*}}))$ 。根据 IH，这等于 $\text{Eval}((CV_{\{\Gamma^*\}}(t_1) := CV_{\{\Gamma^*\}}(t_2)))$ 。

句法典范值: 因 $(t_1 \sim CV_{\{\Gamma\}}(t_1)) \in \Gamma$ 且 $(t_2 \sim CV_{\{\Gamma\}}(t_2)) \in \Gamma$ ，通过

(Congruence)， $((t_1 := t_2) \sim (CV_{\{\Gamma\}}(t_1) := CV_{\{\Gamma\}}(t_2))) \in \Gamma$ 。因此 $CV_{\{\Gamma\}}(t)$ 必是 $(CV_{\{\Gamma\}}(t_1) := CV_{\{\Gamma^*\}}(t_2))$ 的范式，即 $CV_{\{\Gamma\}}(t) = \text{Eval}((CV_{\{\Gamma\}}(t_1) := CV_{\{\Gamma\}}(t_2)))$ 。

比较: 语义求值结果与句法典范值完全相同。

归纳得证。

证毕。

定理 10.10 (强完备性)

若 $\Gamma \models t_1 \sim t_2$ ，则 $\Gamma \vdash t_1 \sim t_2$ 。

- **证明:**

我们证明其逆否命题：若 $\Gamma \not\models t_1 \sim t_2$ ，则 $\Gamma \not\vdash t_1 \sim t_2$ 。

1. **假设:** $\Gamma \not\models t_1 \sim t_2$ 。

2. **建立一致性:** 此假设意味着 Γ 必须是一致的。若 Γ 不一致，则 $\Gamma \vdash \perp \sim ()$ ，由此可证明任何判断，包括 $\Gamma \vdash t_1 \sim t_2$ ，但这与我们的假设矛盾。

3. **构造 Henkin 集:** 由于 Γ 是一致的，根据**Henkin-Lindenbaum 引理 (定理 10.6)**，存在一个 Henkin 集 Γ^* 使得 $\Gamma \subseteq \Gamma^*$ 且 $(t_1 \sim t_2) \notin \Gamma^*$ (因为 Γ^* 也是一致的，且 Γ^* 演绎闭包)。

4. **构造典范模型:** 我们根据**定义 10.8** 构造典范模型 M_{Γ^*} 。

5. **验证模型:**

- M_{Γ^*} 满足 Γ : 取任意 $(t_a \sim t_b) \in \Gamma$ 。因 $\Gamma \subseteq \Gamma^*$ ，则 $(t_a \sim t_b) \in \Gamma^*$ 。根据**推论 10.7**的唯一性， $CV_{\{\Gamma^*\}}(t_a) \equiv CV_{\{\Gamma^*\}}(t_b)$ 。根据**真理引理 (10.9)**， $\llbracket t_a \rrbracket_{M_{\Gamma^*}} = CV_{\{\Gamma^*\}}(t_a)$ 且 $\llbracket t_b \rrbracket_{M_{\Gamma^*}} = CV_{\{\Gamma^*\}}(t_b)$ 。

因此 $\llbracket t_a \rrbracket_{M_{\Gamma^*}} = \llbracket t_b \rrbracket_{M_{\Gamma^*}}$ ，即 $M_{\Gamma^*} \models t_a \sim t_b$ 。故 $M_{\Gamma^*} \models \Gamma$ 。

- M_{Γ^*} 不满足 $t_1 \sim t_2$ ：我们已知 $(t_1 \sim t_2) \notin \Gamma^*$ 。根据**推论 10.7**，这意味着 $CV_{\{\Gamma^*\}}(t_1) <\text{binary data, 2 bytes}> CV_{\{\Gamma^*\}}(t_2)$ 。根据**真理引理**， $\llbracket t_1 \rrbracket_{M_{\Gamma^*}} = CV_{\{\Gamma^*\}}(t_1)$ 且 $\llbracket t_2 \rrbracket_{M_{\Gamma^*}} = CV_{\{\Gamma^*\}}(t_2)$ 。因此， $\llbracket t_1 \rrbracket_{M_{\Gamma^*}} \neq \llbracket t_2 \rrbracket_{M_{\Gamma^*}}$ ，即 $M_{\Gamma^*} \not\models t_1 \sim t_2$ 。

6. **结论**：我们构造了一个模型 M_{Γ^*} ，它满足 Γ 但不满足 $t_1 \sim t_2$ 。根据语义蕴含的定义，这证明了 $\Gamma \not\models t_1 \sim t_2$ 。

逆否命题得证，强完备性定理成立。

证毕。