arXiv:2302.03305v1 [cs.LO] 7 Feb 2023

通过动态逻辑定性表示不确定性与论证动态*

Antonio Yuste-Ginel[1] 和 Andreas Herzig[1]

[1]IRIT，法国国家科学研究中心，法国图卢兹，
herzig@irit.fr，antonioyusteginel@gmail.com

摘要

动态与不确定性是现实生活中论证的重要特征。近期许多研究致力于将这两个方面结合起来，并纳入 Dung 著名的抽象论证框架（论证框架）中。本文提出通过一种良好的逻辑工具——命题赋值动态逻辑（DL-PA）——结合这两条研究路线。我们的研究结果表明，几乎所有现有形式体系中用于定性表示论证框架不确定性的主要推理任务，都可以在 DL-PA 中编码。此外，这一工具也适用于刻画动态结构，例如控制论证框架，并用于在不确定性下发展更精细的论证性交流形式。

1 引言

形式论证已被证明是非单调推理以及诸多其他应用中的一种成功方法 [17, 4, 55]。在旨在为基于论证的推理提供形式模型的研究中，论证的抽象模型发挥着至关重要的作用，因为它们回答了一个相当根本的问题：理性主体应如何从一组相互冲突的论证中选择那些得到更充分论证的论证？“抽象”这一形容词强调，这些模型忽略论证的性质和结构，以便专注于不同的语义；借助这些语义，可以对上述问题给出精确回答。论证最主要的抽象模型是使用有向图，这是Dung 在 [39] 中首次提出的，并以论证框架（论证框架）的名称出现；其中，节点代表论证，箭头代表论证之间的攻击。

*这是论文“Antonio Yuste-Ginel 和 Andreas Herzig，《通过动态逻辑研究定性不确定性与论证动态》，*Journal ofLogic and Computation*，2023年；exac098，https://doi.org/10.1093/logcom/exac098”的初步版本。请查阅原始出版物，并在引用时使用完整的参考文献。

尽管论证框架是一种优雅而强大的工具，但对于许多用途而言，其建模能力仍过于有限。因此，文献中提出了许多对Dung模型的扩展，其中最突出的是支持关系 [25], 、递归形式的攻击 [8], 以及论证之间的偏好关系 [3]。所有这些方法都有两个关键局限：（i）它们具有静态特征；（ii）假设形式化智能体对论证框架的结构拥有完美知识，也就是说，对辩论中的相关论证和攻击拥有完美知识。

关于（i），论证框架可以被理解为一场辩论的快照。事实证明，这种理解有助于为许多有趣的论证概念提供数学上精确的对应形式。然而，论证的一个基本方面在于其动态特征，因为在论证对话的发展过程中，论证、论证之间的冲突以及参与者的观点通常都会发生变化。因此，形式论证系统的动态成为形式论证领域近期研究方向关注的焦点，也就不足为奇了；该方向尤其关注抽象模型。近期综述可参见[38] 和 [11] 。

至于（ii），在对抗性情境中，这一假设会成为一个显著的缺陷，因为人们通常希望对主体认为其对手所持有的信息（即论证框架的一部分）进行建模，因此不确定性便自然产生。通过研究能够涵盖不同形式不确定性的论证框架扩展，这一完全知识假设已得到放宽；不确定性既可以是概率性的，也可以是定性的。有关这两种方法的近期综述，分别参见 [51] 和 [52] 。在后一类方法中，不完全论证框架（IAFs） [13, 40, 14, 16, 15] 和控制论证框架（CAFs） [31, 58, 32] 近期受到广泛关注，由此形成了对相关推理任务复杂性的精确图谱，并涌现出一些应用 [32]。

与此同时，形式论证领域的大量研究致力于构建适用于论证形式体系推理的逻辑理论，尤其关注论证框架及其动态；近期综述见 [18] 。命题赋值动态逻辑（DL-PA） [6] 已被证明是开展这项工作的有效工具 [35, 37, 36, 48]。DL-PA 是命题动态逻辑（PDL） [46], 的一个性质良好的变体，其中原子程序仅限于将命题变量赋值为真或假。它的表达能力足以刻画所有标准的论证语义。与命题逻辑中的编码相比，DL-PA 能够以更简洁的方式刻画包含极小性或极大性准则的语义。此外，研究还强调了它相较于量化布尔公式等同样简洁的语言所具有的优势 [36]。

本研究进一步推进了在 DL-PA 中对抽象论证框架进行逻辑编码，致力于实现三个总体目标：（1）刻画此前尚未得到刻画的论证语义，其中一些语义需要采用颇具挑战性的编码方法；（2）通过将不同论证框架扩展中的推理任务归约为 DL-PA模型检验问题，在 DL-PA 中整合有关论证框架的定性不确定性以及论证动态；（3）通过开发

受我们的编码直接启发的、在不确定性下对论证性交流进行建模的新形式，从而表明所选逻辑也是一种适合探索性研究的工具。

在第 2 节介绍论证框架和 DL-PA 的必要背景之后，第 3 节给出了 DL-PA 中多种论证框架语义的多项式编码。第 4 节介绍了若干用于定性表示论证框架相关不确定性的形式体系，并将其主要推理任务归约为 DL-PA 模型检测问题。第 5 节讨论了论证框架的动态与定性不确定性的联合方法。第 6 节以对未来工作的讨论和挑战为全文作结。证明和证明概要见附录。

2 背景

全文中，我们假设论证的全域$\mathcal{U}$ 是一个固定、有限且非空的集合。此外，我们假设 $\mathcal{U}$ 足够大，能够容纳我们的示例。论证集合（记作 A，有时带上标）被视为 $\mathcal{U}$的子集；所有冲突关系（记作 R，有时带上标）都是 $\mathcal{U}$上的二元关系，即 $R \subseteq \mathcal{U} \times \mathcal{U}$。给定 $A \subseteq \mathcal{U}$ 和$R \subseteq \mathcal{U} \times \mathcal{U}$，我们用 $R \restriction_A$ 来简写 $R \cap (A \times A)$（即将 R 限制在A上）。

2.1 抽象论证框架（论证框架）及其语义

论证框架（论证框架）是一个有向图（A, R） [39], ，其中 A表示论证的集合，R 表示它们之间基于冲突的关系（通常是攻击关系）。[1] 我们将 $\mathcal{AF}$ 记为所有论证框架（基于 $\mathcal{U}$）的集合。论证语义旨在刻画辩论中合理立场这一非形式化概念。文献中包含大量此类语义。它们通常以基于扩展或基于标记的术语来呈现。对于大多数现有语义，已证明这两种方法（扩展和标记）等价。本文采用基于扩展的呈现方式，并将讨论范围限制在少数几种语义；感兴趣的读者可参阅 [7] 了解概述。

首先定义一些有用的概念。设 (A, R) 是一个论证框架，且有 $E \subseteq A$。我们定义 $E^+ = \{x \in A \mid \exists y \in E : (y, x) \in R\}$ （被 E 攻击的论证集合），以及 $E^\oplus = E \cup E^+$ （所谓的 E 范围）。论证集合 $E \subseteq A$ 是无冲突的，当且仅当 $E \cap E^+ = \emptyset$。此外， E 防御 $a \in A$，当且仅当对于每个 $x \in A$：如果 $(x, a) \in R$，那么 $x \in E^+$。最后，$E \subseteq A$ 是可接受的，当且仅当它 (i) 无冲突，且 (ii) 自我防御（它防御其所有成员）。

[1]与 $A \subseteq \mathcal{U}$一样，我们实际上聚焦于 $finite$ 论证框架，因为大多数文献也是如此。这是我们方法固有的局限：我们的编码需要对 $\mathcal{U}$进行量化，因此 $\mathcal{U}$ 必须是有限的。事实证明，要刻画一些更一般的论证语义，需要强大的逻辑语言，例如用于基础语义的模态 μ-演算 [43]。

在 [39], , Dung 引入了四种不同的语义。论证集合 $E \subseteq A$被称为:

- 稳定扩展当且仅当（i）它是无冲突的，并且（ii）$A \backslash E \subseteq E^{+}$ （“E 攻击其自身之外的每个论证”）；
- 完全扩展当且仅当（i）它是无冲突的；并且（ii）它恰好包含 A 其所防御的论证；
- 基础扩展当且仅当它是一个极小（相对于集合包含关系）的完全扩展；
- 优选扩展当且仅当它是一个极大（相对于集合包含关系）的完全扩展。

众所周知，对于任意论证框架，完全、基础和优选扩展的存在性都有保证。然而，稳定语义并非如此：有些框架不存在稳定扩展。此外，在上述语义中，基础语义是唯一属于所谓单一状态方法的语义：每个论证框架恰好有一个基础扩展。当我们将主体的信念建模为其论证评估过程的输出时，这一点便构成了一项优势。更确切地说，如果某种语义允许论证框架存在多个扩展，那么在采用结构化论证时，这些扩展在逻辑上通常互不相容，而且没有明确的方法可以从中作出选择。

除了上述 Dung 的四种语义外，我们还会考虑其他一些语义。

半稳定语义的提出，是为了解决某些条件下稳定扩展缺失所带来的问题。当且仅当 E 是一个范围最大的（相对于集合包含关系）完全扩展时，集合 $E \subseteq A$ 才是 (A, R) 的一个半稳定扩展。更正式地说， E 是一个半稳定扩展，当且仅当

(i) E 是一个完全扩展；并且

(ii) 不存在其他完全扩展 E' 使得 $E^{\oplus} \subset E'^{\oplus}$。

与稳定扩展的情况相反，每个有限论证框架中至少存在一个半稳定扩展。此外，当稳定扩展集合非空时，稳定扩展与半稳定扩展相同 [23]。

尽管基 grounded 语义因采用单状态方法而颇具吸引力，但它也会因过于怀疑而受到批评，因为它通常会留下许多未决论证，即既不属于基 grounded 扩展、也不受其攻击的论证。积极语义和理想语义的理念，是在保留返回单一扩展这一优点的同时，避免过度怀疑。

形式上，集合 $E \subseteq A$ 是（A, R）的理想集，当且仅当它是可接受的，且包含于每个优选扩展中。（A, R）的理想扩展是其极大的（相对于集合包含关系）理想集。

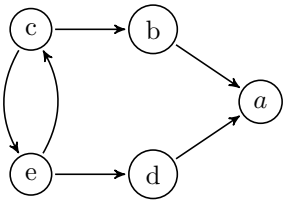
此外，集合 $E \subseteq A$ 是急切集，当且仅当它是可接受的，且包含于每个半稳定扩展中。（A, R）的急切扩展是其极大的（相对于集合包含关系）急切集。

上述所有语义都满足所谓的可接受性原则，即它们的所有扩展都是可接受集。然而，在某些情况下，自我辩护可能是过于严格的要求，例如，在刻画人类论证评估时 [45]。另一类用于选取特定无冲突集的语义被归为基于天真性的语义（例如参见 [30]）。这些语义都以天真扩展这一概念为基础。(A, R) 的天真扩展就是一个极大（相对于集合包含关系）的无冲突集。一种更精细的、深受半稳定扩展概念启发的基于天真性的语义是阶段语义。形式上，(A, R) 的阶段扩展是在所有无冲突集中范围极大的无冲突集。

我们以显而易见的方式使用简写$\{st, co, gr, pr, se, id, ea, na, stg\}$来缩写各语义的名称。对于每个 $\sigma \in \{st, co, gr, pr, se, id, ea, na, stg\}$，我们将 (A, R) 的所有 σ-扩展构成的集合记为 $\sigma(A, R)$。如果论证 $x \in A$ 属于至少一个（分别地，每一个）σ-扩展，则称其在 σ 下是可宽容接受的（分别地，怀疑接受的）。

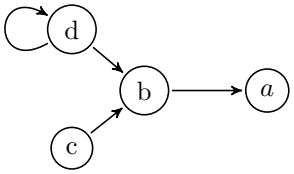
例如，对于下图所示的论证框架 (A_0, R_0)，我们有 $\mathsf{st}(A_0, R_0) = \mathsf{pr}(A_0, R_0) = \mathsf{se}(A_0, R_0) = \mathsf{stg}(A_0, R_0) = \{\{b, e\}, \{c, d\}\}$；$\mathsf{gr}(A_0, R_0) = \mathsf{id}(A_0, R_0) = \mathsf{ea}(A_0, R_0) = \{\emptyset\}$；$\mathsf{co}(A_0, R_0) = \{\emptyset, \{b, e\}, \{c, d\}\}$ 以及 $\mathsf{na}(A_0, R_0) = \{\{a, c\}, \{a, e\}, \{b, e\}, \{b, d\}, \{c, d\}\}$。

(A_0, R_0)



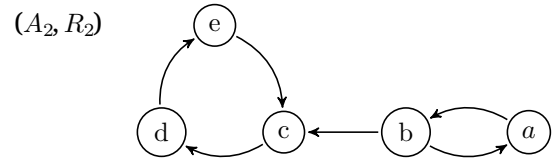
此外，下图所示并引自 [23] 的 (A_1, R_1) 展示了稳定语义与半稳定语义之间的差异：该框架没有稳定扩展，而 $\{c, a\}$ 是一个半稳定扩展（事实上也是唯一的一个）。

(A_1, R_1)



最后，要了解（半）稳定语义与优选语义之间的区别，可以考虑取自 [7] 并如下图所示的 (A_2, R_2)。集合 $\{a\}$ 是一个优选扩展，但它不是（半）稳定扩展。此外，该例还说明了理想语义与急切语义之间的区别，因为急切

扩展是 $\{b, d\}$，而理想扩展为空。更多示例请参阅 [7], ，以及在线图形求解器 ConArg [20]。



2.2 命题赋值动态逻辑（DL-PA）

本文采用 DL-PA 作为通用逻辑框架。DL-PA 的语言由一个可数无限的命题变量集 Prp $= \{p_1, p_2, \ldots\}$构成。我们假设 Prp 包含若干种命题变量，用于表示论证的状态以及它们之间的关系。首先，对于每个论证集 $A \subseteq \mathcal{U}$，我们关联意识变量集 $\mathsf{AW}_A = \{\mathsf{aw}_x \mid x \in A\}$ 和接受变量集 $\mathsf{IN}_A = \{\mathsf{in}_x \mid x \in A\}$。其次，对于每个关系 $R \subseteq \mathcal{U} \times \mathcal{U}$，我们关联攻击变量集 $\mathsf{ATT}_R = \{\mathsf{r}_{x,\ y} \mid (x, y) \in R\}$。因此，我们的逻辑中的命题变量集包含

$$\mathtt{Prp}_{\mathcal{U}} = \mathsf{AW}_{\mathcal{U}} \cup \mathsf{IN}_{\mathcal{U}} \cup \mathsf{ATT}_{\mathcal{U}\times\mathcal{U}}.$$

由于 Prp$_{\mathcal{U}}$ 是有限的，可数无限集 Prp 提供了一个辅助变量储备，可用于帮助我们编码例如半稳定语义和阶段语义。DL-PA 的公式和程序通过互递归定义：

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid [\pi]\varphi,\\ \pi &::= +p \mid -p \mid \varphi? \mid (\pi;\pi) \mid (\pi \cup \pi) \mid \pi^{\smile},\end{aligned}$$

其中 p 遍历 Prp。公式 $[\pi]\varphi$ 的意思是“φ 在每次可能执行 π 后都为真”。程序 $+p$ 使 p 为真，而 $-p$ 使 p 为假。程序 $\varphi?$ 用于测试 φ 是否为真；若为假，则执行失败。程序 $\pi_1;\pi_2$ 是 π_1 和 π_2的顺序组合；而 $\pi_1 \cup \pi_2$ 是它们的非确定性组合。最后，$\pi^{\smile}$ 表示以“相反的方式”执行 π；例如，程序 $+p^{\smile}$ 撤销将 p 赋为真的操作：当 p 为假时，执行失败；当 p 为真时，则非确定性地选择什么也不做，或使 p 为假。

下面再看一些例子。公式 $[-p]\neg p$ 将是有效的：$-p$只有一种执行方式，而且执行后 p 为假。相比之下，$[+p]\neg p$ 将不可满足。此外，对于语法形式不同的 p 和 q，$[+p]q$ 与 q 等价。公式 $[\varphi?]\psi$ 表示，在测试 $\varphi?$ 的每一种可能执行之后，ψ 都为真。此类执行至多只有一种，即 φ 为真时；而且它不会改变任何东西；当 φ 为假时，测试失败，$[\varphi?]\psi$ 便空真。因此，$[\varphi?]\psi$ 必须等价于 $\neg\varphi \vee \psi$。公式 $[\pi_1;\pi_2]\varphi$ 等价于 $[\pi_1][\pi_2]\varphi$，而 $[\pi_1 \cup \pi_2]\varphi$ 将等价于 $[\pi_1]\varphi \wedge [\pi_2]\varphi$。最后，公式 $[+p \cup -p]\neg p$ 和 $[-p^{\smile}]p$ 都将

不可满足。前者之所以如此，是因为存在一种非确定性选择（即选择 $+p$），在此之后 p 为真；后者之所以如此，是因为存在一次非确定性 $-p^\smile$ 的执行，在此之后 p 仍为假。

以下是本文后续部分会用到的一些公式和程序的缩写。程序 $\top$?缩写为 skip：它总是成功，且不会改变任何东西。程序（$\varphi?;\ \alpha$） $\cup$（$\neg\varphi?;\ \beta$）缩写为 if φ then α else β。后一种写法的一个特例是 β 为 skip，此时我们只写 if φ then π。（注意，这与 $\varphi?;\ \alpha$不同：当 φ为假时，后者会失败，而前者会成功且不做任何事。）至于公式，缺少的布尔联结词按通常方式定义。此外，公式 $\langle\pi\rangle\varphi$ 是 $\neg[\pi]\neg\varphi$的缩写。因此，它的含义是“经过 π的一次可能执行后，φ 为真”。特别地，$\langle\pi\rangle\top$ 应读作“π 可执行”。例如，公式 $\varphi \to [\pi]\langle\pi^\smile\rangle\varphi$ 表示 π 的每次成功执行都可以被逆转；该公式将是有效的。（注意，不能用方框替换菱形，正如无效公式 $p \to [+p][+p^\smile]p$所示。）

我们的模型是 Prp 上的经典命题赋值，即 Prp 的子集。我们用 v、v ′、v ″ 表示赋值。公式 φ 的解释方式与动态逻辑类似，而程序 π 则被解释为赋值上的二元关系。与语法一样，DL-PA 的语义通过互递归来定义。公式的解释如下：

$$\begin{array}{ll} v \models p & \text{if} \quad p \in v, \\ v \models [\pi]\varphi & \text{if} \quad (v, v') \in ||\pi|| \text{ implies } v' \models \varphi, \end{array}$$

布尔联结词的解释与通常相同；程序的解释如下：

$$\begin{aligned} ||+p|| &= \{(v, v') \mid v' = v \cup \{p\}\}, \\ ||-p|| &= \{(v, v') \mid v' = v \setminus \{p\}\}, \\ ||\varphi?|| &= \{(v, v) \mid v \models \varphi\}, \\ ||\pi; \pi'|| &= ||\pi|| \circ ||\pi'||, \\ ||\pi \cup \pi'|| &= ||\pi|| \cup ||\pi'||, \\ ||\pi^\smile|| &= ||\pi||^{-1}. \end{aligned}$$

$+p$ 的解释是使 p 为真、同时不改变其他任何内容的关系；$-p$ 的解释也类似。测试 φ? 的解释将所有使 φ 为真的赋值与其自身关联；例如，$||\top?|| = \{(\mathrm{v}, \mathrm{v}) \mid \mathrm{v} \subseteq \mathrm{Prp}\}$。顺序组合 $\pi_1;\pi_2$ 自然地解释为关系复合，而非确定性组合 $\pi_1 \cup \pi_2$ 则解释为两个关系 $||\pi_1||$ 和 $||\pi_2||$的集合并。逆向程序 $\pi^\smile$ 的解释是关系 $||\pi||$ 的逆，它将赋值 v 与所有满足 π 可执行且可能导向 v 的赋值关联起来。例如，$||+p^\smile|| = \{(\mathrm{v}', \mathrm{v}) \mid \mathrm{v}' = \mathrm{v} \cup \{p\}\}$。

如果存在某个 v 使得 $\mathrm{v} \models \varphi$，则公式 φ 是 DL-PA 可满足的；如果对每个 v 都有 $\mathrm{v} \models \varphi$，则它是 DL-PA 有效的。例如，$\langle+p\rangle\top$ 和 $\langle-p\rangle\top$ 都有效，而$\langle-p^\smile\rangle p$ 和 $\langle-p^\smile\rangle\neg p$ 可满足但无效。众所周知，可满足性、有效性和模型检测都是 PSPACE 完全的判定问题 [5]。

现在我们来介绍一些稍后会用到的 DL-PA 程序。设 $\mathsf{P} = \{p_1, \ldots, p_n\} \subseteq \mathsf{Prp}$ 是一个命题变量的有限集。首先，我们定义

$$;_{p\in\mathsf{P}}\pi_p = \pi_{p_1}; \ldots; \pi_{p_n}.$$

按照惯例，我们假设当 $\mathsf{P} = \emptyset$ 时，该缩写等同于 skip。对于非确定性并，我们也采用相同的惯例，即 $\bigcup_{p\in\emptyset}\pi_p = \mathsf{skip}$。原则上，$\mathsf{P}$ 中各元素的顺序很重要，但每次使用这种记法时，我们都会确保相关程序 π_{p_i} 不受顺序影响。以下缩写尤其满足这一点：

$$\begin{aligned}
\mathsf{mkTrueOne}(\mathsf{P}) &= \bigcup_{p\in\mathsf{P}}(\neg p?; +p) = (\neg p_1?; +p_1) \cup \ldots \cup (\neg p_n?; +p_n),\\
\mathsf{mkFalseOne}(\mathsf{P}) &= \bigcup_{p\in\mathsf{P}}(p?; -p) = (p_1?; -p_1) \cup \ldots \cup (p_n?; -p_n),\\
\mathsf{mkTrueSome}(\mathsf{P}) &= ;_{p\in\mathsf{P}}(+p \cup \mathsf{skip}) = (+p_1 \cup \mathsf{skip}); \ldots; (+p_n \cup \mathsf{skip}),\\
\mathsf{mkFalseSome}(\mathsf{P}) &= ;_{p\in\mathsf{P}}(-p \cup \mathsf{skip}) = (-p_1 \cup \mathsf{skip}); \ldots; (-p_n \cup \mathsf{skip}),\\
\mathsf{vary}(\mathsf{P}) &= ;_{p\in\mathsf{P}}(+p \cup -p) = \big(+p_1 \cup -p_1\big); \ldots; \big(+p_n \cup -p_n\big).
\end{aligned}$$

程序 mkTrueOne(P) 从 P 中选择一个元素，检查它是否为假，并将其设为真；而 mkTrueSome(P) 则将 P 中先前为假的某些元素设为真。程序 mkTrueOne(P) 和 mkFalseOne(P) 互为逆操作；mkTrueSome(P) 和 mkFalseSome(P) 也一样。顺序组合 mkTrueOne(P); mkTrueSome(P) 会将 P 中至少一个先前为假的元素设为真（也可能设为多个）。最后一个程序——即 vary(P)——与顺序组合 mkTrueSome(P); mkFalseSome(P) 和 mkFalseSome(P); mkTrueSome(P) 的含义相同。

我们正式表述这些程序的含义：[2]

命题 1. 我们有：

$$\begin{aligned}
||\mathsf{mkTrueOne}(\mathsf{P})|| &= \{(v, v') \mid v' = v \cup \{p\} \textit{ for some } p \in \mathsf{P}\backslash v\},\\
||\mathsf{mkFalseOne}(\mathsf{P})|| &= \{(v, v') \mid v' = v \setminus \{p\} \textit{ for some } p \in \mathsf{P}\cap v\},\\
||\mathsf{mkTrueSome}(\mathsf{P})|| &= \{(v, v') \mid v' = v \cup \mathsf{P}' \textit{ for some } \mathsf{P}' \subseteq \mathsf{P}\},\\
||\mathsf{mkFalseSome}(\mathsf{P})|| &= \{(v, v') \mid v' = v \setminus \mathsf{P}' \textit{ for some } \mathsf{P}' \subseteq \mathsf{P}\},\\
||\mathsf{vary}(\mathsf{P})|| &= \{(v, v') \mid v \setminus v' \subseteq \mathsf{P} \textit{ and } v' \setminus v \subseteq \mathsf{P}\}.
\end{aligned}$$

从赋值到论证框架，再反向转换。由于我们假设 Prp 包含 $\mathsf{Prp}_{\mathcal{U}}$，每个赋值 $\mathsf{v} \subseteq \mathsf{Prp}$ 都表示由下式定义的论证框架 (A_v, R_v)：

$$\begin{aligned}
A_v &= \{x \in \mathcal{U} \mid \mathtt{aw}_x \in v\},\\
R_v &= \{(x, y) \in \mathcal{U} \times \mathcal{U} \mid \mathtt{r}_{x,y} \in v\} \restriction_{A_v}\\
&= \{(x, y) \in A_v \times A_v \mid \mathtt{r}_{x,y} \in v\}.
\end{aligned}$$

[2]以下命题是对 [36, 引理 1]的略微修正。

反过来，每个论证框架（A， R）都由赋值表示

$$v_{(A,R)} = \{\mathtt{aw}_x \mid x \in A\} \cup \{\mathtt{r}_{x,y} \mid (x,y) \in R\}.$$

注意，对于任意集合 $A \subseteq \mathcal{U}$ 和关系$R \subseteq \mathcal{U}\times\mathcal{U}$，赋值 $\mathsf{v}^{(A,R)}$ 都是良好定义的，即使 (A, R) 不是论证框架。（只要 R 中包含不属于 $A\times A$ 的对 $(x, y) \in \mathcal{U}\times\mathcal{U}$，就会出现这种情况。）此外，请注意，如果我们从赋值 v' 出发，那么 $\mathsf{v}_{(A_{v'},R_{v'})} = \mathsf{v}'$ 通常并不成立，因为赋值可以包含攻击变量 $\mathsf{r}_{a,\ b}$，却不包含 aw_a 和 aw_b。不过，如果我们从论证框架 (A', R') 出发，那么 $(A_{v_{(A',R')}}, R_{v_{(A',R')}}) = (A', R')$ 总是成立。最后，对于每个赋值 v，我们将与 v 相关联的扩展定义为：

$$\mathsf{E}_v = \{x \in \mathcal{U} \mid \mathtt{in}_x \in v\}.$$

3 DL-PA 中的论证语义

现在我们说明如何在 DL-PA 中刻画论证语义。首先，采用 [19] 中介绍的、用命题逻辑编码论证框架的方法。具体而言，对每种语义 σ，关联一个公式 φ_σ，使得 $\mathsf{v} \models \varphi_\sigma$ 当且仅当 E_v 是 (A_v, R_v) 的一个 σ-扩展。[35, 37, 36, 48]，进一步发展了这一方法，提出不应局限于刻画扩展，而应利用 DL-PA 程序描述扩展的计算过程。最基本的方法是采用“生成与检验”的思路：通用程序

$$\mathsf{makeExt}^\sigma = \mathsf{vary}(\mathsf{IN}_\mathcal{U}); \varphi_\sigma?$$

通过先改变接受变量的取值，再检查是否得到一个 σ-赋值，以非确定性方式构造所有可能的 σ-扩展。正如 [36], 所阐明的那样，其他更高效的扩展构造算法也可以表示为 DL-PA 程序，并且可以证明它们与 $\mathsf{makeExt}^\sigma$等价。

由于我们假设论据背景全集为 $\mathcal{U}$，因此需要一种将意识变量 aw_x 纳入考量的论证语义编码。[37] 已针对稳定语义完成了这项工作。[3] 在此，我们将该编码扩展至第2.1节介绍的其余语义。所有编码的正确性将在本节末尾正式阐述。

我们首先定义一些公式，以便用紧凑的方式表示不同的语义。

3.1 实用的 DL-PA 公式

以下 DL-PA 公式表达了：由接受变量标识的论证，确实是形式化代理所考虑的论证

[3]在 [37], 中，使用术语 *enablement* 和记号 En_x，而非 *awareness* 和 $\mathtt{aw}_x$。

（她意识到的论证）：

$$\mathsf{Well} = \bigwedge_{x \in \mathcal{U}} (\mathtt{in}_x \to \mathtt{aw}_x).$$

这个缩写使我们能够表达无冲突性和可接受性：

$$\mathsf{ConFree} = \mathsf{Well} \wedge \bigwedge_{x \in \mathcal{U}} \bigwedge_{y \in \mathcal{U}} \neg(\mathtt{in}_x \wedge \mathtt{in}_y \wedge \mathtt{r}_{x,y}),$$

$$\mathsf{Admissible} = \mathsf{ConFree} \wedge \bigwedge_{x \in \mathcal{U}} \Big(\mathtt{in}_x \to \bigwedge_{y \in \mathcal{U}} \big((\mathtt{aw}_y \wedge \mathtt{r}_{y,x}) \to \bigvee_{z \in \mathcal{U}} (\mathtt{in}_z \wedge \mathtt{r}_{z,y})\big)\Big).$$

我们对半稳定扩展和阶段扩展的刻画使用了$'_x$ 中变量x的新副本，每个 $x \in \mathcal{U}$ 对应一个副本（这是可行的，因为 Prp 是可数无限集，而 $\mathcal{U}$ 是有限集）。对于这些辅助变量，我们定义了一个程序，用于复制 $\mathsf{IN}_{\mathcal{U}}$ 变量的取值：

$$\mathsf{copy}(\mathsf{IN}_{\mathcal{U}}) = \mathbin{;}_{x \in \mathcal{U}} ((\mathtt{in}_x?; +\mathtt{in}'_x) \cup (\neg\mathtt{in}_x?; -\mathtt{in}'_x)).$$

此外，以下两个公式刻画了由 v 表示的扩展 E_v（记作 $\mathrm{E}_v^{\oplus}$）的范围是否包含于由这些副本表示的扩展的范围中；反之亦然：

$$\mathsf{IncludedInCp} = \bigwedge_{x \in \mathcal{U}} \left[\left(\mathtt{in}_x \vee \left(\mathtt{aw}_x \wedge \bigvee_{y \in \mathcal{U}} (\mathtt{in}_y \wedge \mathtt{r}_{y,x}) \right) \right) \to \left(\mathtt{in}'_x \vee \left(\mathtt{aw}_x \wedge \bigvee_{y \in \mathcal{U}} (\mathtt{in}'_y \wedge \mathtt{r}_{y,x}) \right) \right) \right],$$

$$\mathsf{IncludesCp} = \bigwedge_{x \in \mathcal{U}} \left[\left(\mathtt{in}'_x \vee \left(\mathtt{aw}_x \wedge \bigvee_{y \in \mathcal{U}} (\mathtt{in}'_y \wedge \mathtt{r}_{y,x}) \right) \right) \to \left(\mathtt{in}_x \vee \left(\mathtt{aw}_x \wedge \bigvee_{y \in \mathcal{U}} (\mathtt{in}_y \wedge \mathtt{r}_{y,x}) \right) \right) \right].$$

最后，为了刻画理想语义和积极语义，我们需要确保所考虑的集合是可接受的，并且属于每个优选扩展（对于理想语义），或每个半稳定扩展（对于积极语义）。借助扩展构造程序 $\mathrm{makeExt}^{\sigma}$，可以简洁地实现这一点：

$$\mathsf{IdealSet} = \mathsf{Admissible} \wedge \bigwedge_{x \in \mathcal{U}} (\mathtt{in}_x \to [\mathsf{makeExt}^{pr}]\mathtt{in}_x),$$

$$\mathsf{EagerSet} = \mathsf{Admissible} \wedge \bigwedge_{x \in \mathcal{U}} (\mathtt{in}_x \to [\mathsf{makeExt}^{se}]\mathtt{in}_x).$$

稳定 $=$ 良好 $\wedge \bigwedge_{x\in\mathcal{U}} \Big(\mathrm{aw}_x \rightarrow (\mathrm{in}_x \leftrightarrow \neg \bigvee (\mathrm{in}_y \wedge \mathrm{r}_{y,x})\Big)$, 完全 $=$ 无冲突 $\wedge \bigwedge_{y\in\mathcal{U}} \Big(\mathrm{in}_x \leftrightarrow \bigwedge_{y\in\mathcal{U}} ((\mathrm{aw}_y \wedge \mathrm{r}_{y,x}) \rightarrow \bigvee_{z\in\mathcal{U}}(\mathrm{in}_z \wedge \mathrm{r}_{z,y}))\Big)$, 基 grounded 语义 $=$ 完全 $\wedge$ [mkFalseOne($\mathrm{IN}_{\mathcal{U}}$); mkFalseSome($\mathrm{IN}_{\mathcal{U}}$)]¬完全, 优选语义 $=$ 可接受 $\wedge$ [mkTrueOne($\mathrm{IN}_{\mathcal{U}}$); mkTrueSome($\mathrm{IN}_{\mathcal{U}}$)]¬ 可接受 , 朴素语义 $=$ 无冲突 $\wedge$ [mkTrueOne($\mathrm{IN}_{\mathcal{U}}$)]¬无冲突, 半稳定 $=$完全 $\wedge$ [copy($\mathrm{IN}_{\mathcal{U}}$); makeExt^{co}] (IncludesCp $\rightarrow$IncludedInCp) , 阶段语义 $=$ 无冲突 $\wedge$[copy($\mathrm{IN}_{\mathcal{U}}$); vary($\mathrm{IN}_{\mathcal{U}}$); 无冲突?] (IncludesCp $\rightarrow$IncludedInCp) , 理想语义 $=$理想集 $\wedge$ [mkTrueOne($\mathrm{IN}_{\mathcal{U}}$); mkTrueSome($\mathrm{IN}_{\mathcal{U}}$)]¬理想集, 积极语义 $=$积极集 $\wedge$ [mkTrueOne($\mathrm{IN}_{\mathcal{U}}$); mkTrueSome($\mathrm{IN}_{\mathcal{U}}$)]¬积极集。

表1：通过 DL-PA 公式对第2.1节的语义进行编码

3.2 在 DL-PA 中对第2.1节的语义进行编码

表1列出了所有编码。稳定语义和完全语义的编码略微简化了 [37, 48]的编码。我们对基 grounded 语义、完全语义和优选语义的编码直接改编了 [36] 中用于计算极小性和极大性准则的编码。前四种编码本质上是 [37] 和 [36], 中提出的编码的结合，并进行了一些小幅改进和调整。在此前尚未用 DL-PA 表示的语义中，我们对朴素语义的编码简化了相对于其他语义（如优选语义）检查集合极大性的程序，因为任何包含冲突的集合的超集都不可能是无冲突的。

定理 1。设 $\sigma \in \{st, co, gr, pr, se, id, ea, na, stg\}$。设 $\mathrm{V} \subseteq \mathrm{Prp}$。设（$A$, R）是一个论证框架。则:

- $v \models \varphi_\sigma$ *iff* $\mathsf{E}_v \in \sigma(A_v, R_v)$;
- $\sigma(A, R) = \{\mathsf{E}_v \mid (v_{(A,R)}, v) \in ||\mathsf{makeExt}^\sigma||\}$.

证明可见附录，其他所有结果的证明或证明概要也都收录于此。

4 通过 DL-PA 研究抽象论证中的定性不确定性

本节回顾用于表示论证框架不确定性的现有形式体系。我们将关注范围限定在定性不确定性形式上，即不使用概率或任何其他数值工具的表示方式。

具体而言，我们将介绍：不完整论证框架 [13], 其扩展版本 [53], 受约束的不完整论证框架 [48, 54],以及带依赖关系的不完整论证框架 [41, 42]。研究这些形式体系的主要动机在于，现实生活中的论证存在多种不确定性来源。例如，论证可能十分复杂，以至于推理主体无法确定是否应将其纳入考量，或它们是否会攻击其他论证。或许更常见的是，当主体推理对手的论证情境时，论证中会出现不确定性。由于无法全面了解对手，主体可能无法确定对方是否考虑了某些论证，或是否察觉到某些攻击。而这对于选择恰当的论证来说至关重要，因为这些论证能够说服对手。本文余下部分将以这一后者的直觉为指引。

本节介绍的所有形式体系都基于同一个思路：通过补全这一概念来表示不确定性。补全是指假设性地消除不确定性，使形式化代理在假定对手的论证框架是某种特定情形的前提下进行推理。用认知逻辑的术语来说，这对应于可能世界的概念，正如 [14, 15], 中所提及、并在[59, 50]中详细研究的那样。若想进一步比较本节介绍的形式体系与认知逻辑，读者可参阅第 6 节。

介绍每种形式体系后，我们会说明如何将相关的主要推理任务归约为DL-PA模型检验问题。最后，我们将比较不同的方法。

4.1 不完整论证框架

不完全论证框架 [13] (IAF) 是一个二元组 IAF $=(F,U)$，其中 $F=(A^F,R^F)$ 称为固定部分，$U=(A^?,R^?)$ 称为不确定部分，$R,R^? \subseteq (A^F \cup A^?) \times (A^F \cup A^?)$，$A^F \cap A^? = \emptyset$ 和 $R^F \cap R^? = \emptyset$。因此，IAF 本质上是一个将论证和攻击划分为两个不相交集合的论证框架。谈及不完全论证框架时，我们有时会省略内部括号，也就是说，我们写作 $(A^F,R^F,A^?,R^?)$，而不是 $((A^F,R^F),(A^?,R^?))$。注意，根据定义，不确定论证之间可以存在固定攻击（有时称为条件确定攻击 [14]）。直观而言，我们可以将其理解为：只要主体认为对手知道相关论证，主体就认为对手会认可这些攻击。

IAF $=(A^F,R^F,A^?,R^?)$ 的一个补全是任何满足以下条件的论证框架 (A^*,R^*)：

- $A^F \subseteq A^* \subseteq A^F \cup A^?$; and
- $R^F \restriction_{A^*} \subseteq R^* \subseteq (R^F \cup R^?) \restriction_{A^*}$.

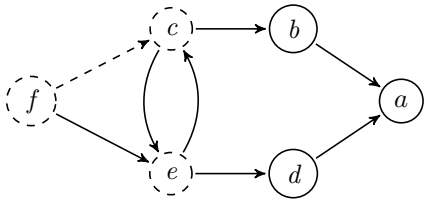
给定一个 IAF IAF，我们将其所有补全的集合记为 completions(IAF)。

一个标准论证框架（A,R）可以视为 IAF（$A,R,\emptyset,\emptyset$），而该 IAF 是其自身唯一的补全。文献中对 IAF 的两个子类已有充分研究，分别是攻击不完全论证框架（简称 att-IAF），[4] 它们是

[4]这个子类此前以 *partial AFs* [24, 28] 之名进行过研究。

具有空 $A^?$ 的 IAF；以及论证不完全论证框架（简称 arg-IAF），它们是具有空 $R^?$ 的 IAF。

例1. 考虑 $\text{IAF}_0 = (A_0^F, R_0^F, A_0^?, R_0^?)$，其中 $A_0^F = \{a, b, d\}$,$R_0^F = \{(b, a)$、(d, a)、(c, b)、(e, d)、(c, e)、(e, c)、$(f, e)\}$，$A_0^? = \{c, e, f\}$ 以及 $R_0^? = \{(f, c)\}$，其图形表示如下。IAF_0 的补全集合如表2所示，但不包括单元格 B2、C2、B4、C4、B5 和 C5。



经典推理任务（如扩展枚举或论证接受）已从论证框架推广到不完全论证框架。这里我们关注如下接受查询：

σ-必要-可质疑接受（σ-NCA）
给定：一个 IAF IAF $= (A^F, R^F, A^?, R^?)$ 以及一个论证 $a \in A^F$。 问题：对于每个（A^*, R^*），是否都存在 $\in$ 补全（IAF） 一个 $E \in \sigma$（A^*, R^*），使得 $a \in E$?

我们可以在上述定义中切换量词，从而得到该问题的不同变体，即可能型和怀疑型变体。请注意，这些推理任务与论证框架中的标准接受问题之间唯一的区别，是增加了一个量化层，即对补全进行量化。

现在，我们的目标是将这些接受问题归约为 DL-PA 模型检测问题。既然我们已经有了用于构造论证框架扩展的程序，这一归约的关键步骤就是设计一个 DL-PA 程序 $\mathsf{makeComp}^{\mathsf{IAF}}$，用于计算 IAF 的所有补全。

首先，IAF 对应的赋值由其固定部分确定：

$$\begin{aligned} v_{\mathsf{IAF}} &= v_{(A^F, R^F)} \\ &= \mathsf{AW}_{A^F} \cup \mathsf{ATT}_{R^F} \\ &= \{\mathtt{aw}_x \mid x \in A^F\} \cup \{\mathtt{r}_{x,y} \mid (x, y) \in R^F\}. \end{aligned}$$

请注意，$(A_{v_{\mathsf{IAF}}}, R_{v_{\mathsf{IAF}}})$ 已经是 IAF 的一个补全：这是最小的补全，其中只考虑固定论据以及它们之间的固定攻击。为了计算 IAF 的所有补全，我们令表示 $A^?$ 中论证以及表示 $R^?$ 中攻击的命题变量子集为真：

$$\mathsf{makeComp}^{\mathsf{IAF}} = \mathsf{mkTrueSome}(\mathsf{AW}_{A^?}); \mathsf{mkTrueSome}(\mathsf{ATT}_{R^?}).$$

下一个命题表明，我们达成了最初的目标。

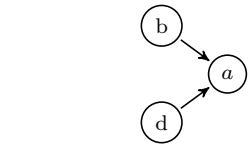
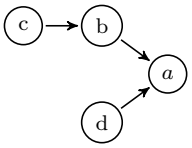
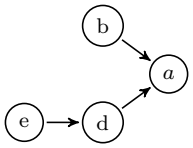
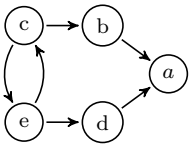
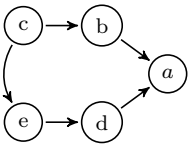
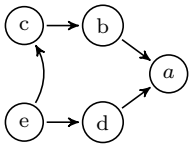
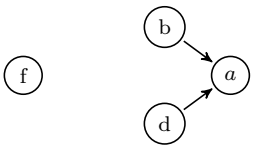
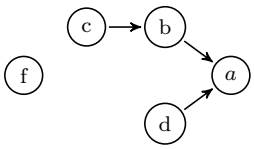
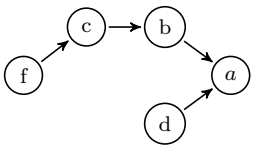
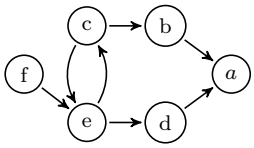
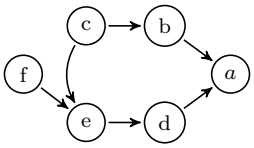
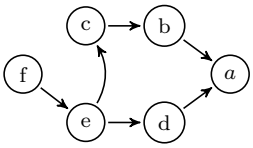
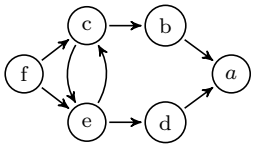
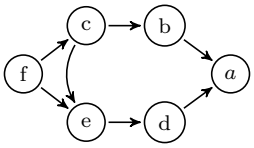
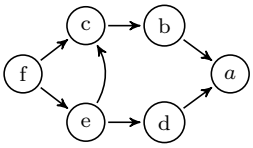
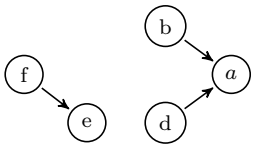
	A	**B**	**C**
1			
2			
3			
4			
5			
6			

表2：CAF_0的补全。列 [**1**, **2**,...、 **6**] 和行 [A、B、C] 仅为编号之用。（空白单元格确实 *not* 表示空补全（∅、∅）。）

命题 2. 设 IAF $= (A^F, R^F, A^?, R^?)$。则：

- 如果 $(\mathbf{v}_{\mathsf{IAF}}, \mathbf{v}) \in ||\mathsf{makeComp}_{\mathsf{IAF}}||$，则 (A_v, R_v) 是 $\in$IAF 的补全。
- 如果 (A^*, R^*) 是 $\in$IAF 的补全，则 $(\mathbf{v}_{\mathsf{IAF}}, \mathbf{v}_{(A^*,R^*)}) \in ||\mathsf{makeComp}_{\mathsf{IAF}}||$。

利用这一结果以及第 3 节提供的计算扩展的一般技术，我们可以将不完全论证框架中的推理问题归约为 DL-PA 中的模型检测问题。

命题 3. 设 IAF $= (F, U)$，$\sigma \in \{st, co, gr, pr, se, id, ea, na, stg\}$，以及$a \in A^F$。则：

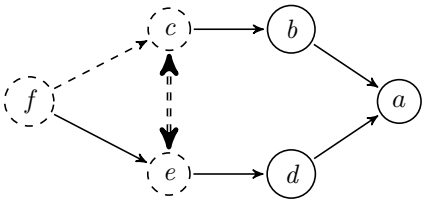
- 对于输入为 IAF 且 a 的 σ-NSA，其答案为“是”当且仅当$v_{\mathsf{IAF}} \models [\mathsf{makeComp}_{\mathsf{IAF}}; \mathsf{makeExt}^{\sigma}]$在$a$中。
- 对于输入为 IAF 和 a 的 σ-NCA，答案为是，当且仅当$v_{\mathsf{IAF}} \models [\mathsf{makeComp}_{\mathsf{IAF}}]\langle \mathsf{makeExt}^{\sigma}\rangle$属于$a$。
- 对于输入为 IAF 和 a 的 σ-PCA，答案为是，当且仅当$v_{\mathsf{IAF}} \models \langle \mathsf{makeComp}_{\mathsf{IAF}};\ \mathsf{makeExt}^{\sigma}\rangle$属于$a$。
- 对于输入为 IAF 和 a 的 σ-PSA，答案为是，当且仅当$v_{\mathsf{IAF}} \models \langle \mathsf{makeComp}_{\mathsf{IAF}}\rangle[\mathsf{makeExt}^{\sigma}]$属于$a$。

4.2 丰富不完整论证框架

丰富不完整论证框架（rIAF） [53] 通过在其不确定部分中添加一种新的（对称且无自反的）不确定关系 $R^{\leftrightarrow} \subseteq (A^F \cup A^?) \times (A^F \cup A^?)$，从而扩展 IAF，$U$ 使得 $R^{\leftrightarrow} \cap R^F = \emptyset$ 且 $R^{\leftrightarrow} \cap R^? = \emptyset$。谈及 rIAF 时，我们有时会省略内部括号，并将其记作 $(A^F, R^F, A^?, R^?, R^{\leftrightarrow})$。新组件 $R^{\leftrightarrow}$ 可以非正式地理解为一个攻击集合，其存在性已知，但方向未知。引入 $R^{\leftrightarrow}$ 的动机是，攻击具有两个基本属性：其存在性和方向。因此，$R^?$ 描述的是关于前者的不确定性，而 $R^{\leftrightarrow}$ 描述的是关于后者的不确定性。请注意，任何 IAF 都可以理解为具有空 $R^{\leftrightarrow}$ 的 rIAF。补全的概念很容易推广到 rIAF，以刻画我们刚才提到的关于 $R^{\leftrightarrow}$ 的直观认识。rIAF $= (A^F, R^F, A^?, R^?, R^{\leftrightarrow})$的一个补全是任意满足以下条件的论证框架 (A^*, R^*)：

- $A^F \subseteq A^* \subseteq (A^F \cup A^?)$;
- $R^F \restriction_{A^*} \subseteq R^* \subseteq (R^F \cup R^? \cup R^{\leftrightarrow}) \restriction_{A^*}$;
- 对于每个 $x, y \in A^*$: $(x, y) \in R^{\leftrightarrow}$ 蕴含 $(x, y) \in R^*$ 或 $(y, x) \in R^*$。

例2. 设 $\mathsf{rIAF}_0 = (A_0^F, A_0^?, R_0^F, R_0^?, R_0^{\leftrightarrow})$，其中 $A_0^F = \{a, b, d\}$，$A_0^? = \{c, e, f\}$，$R_0^F = \{(b, a)、(d, a)、(c, b)、(e, d)、(f, e)\}$，$R_0^? = \{(f, c)\}$，且 $R_0^{\leftrightarrow} = \{(c, e)、(e, c)\}$。我们将 rIAF_0 图形化表示如下：



rIAF_0 的补全集如表2所示。

在 DL-PA 中计算丰富 IAF 的补全会稍微复杂一些，因为程序 mkTrueSome 不足以处理 $R^{\leftrightarrow}$ 的对称攻击。不过，我们可以为此目的定义一个特定程序。

首先，给定 rIAF $=(A^F, R^F, A^?, R^?, R^{\leftrightarrow})$，与 rIAF 相关联的赋值和之前一样由其固定部分决定：

$$\begin{aligned} v_{\mathsf{rIAF}} &= v_{(A^F, R^F)} \\ &= \mathsf{AW}_{A^F} \cup \mathsf{ATT}_{R^F} \\ &= \{\mathtt{aw}_x \mid x \in A^F\} \cup \{\mathtt{r}_{x,y} \mid (x,y) \in R^F\}. \end{aligned}$$

请注意，与不完全论证框架的情况不同，$(A_{v_{\mathsf{rIAF}}}, R_{v_{\mathsf{rIAF}}})$ 并不总是 rIAF 的补全：只要 $R^{\leftrightarrow} \cap (A^F \times A^F)$ 非空，情况就不成立。现在定义一个程序，将 $R^{\leftrightarrow}$ 中的元素整合到每个补全中。

设 $\mathsf{ATT}_R = \{\mathtt{r}_{x_1,\ y_1}, ..., \mathtt{r}_{x_n,\ y_n}\}$ [5] 为一组攻击变量，并定义程序

$$\mathsf{dis}(\mathsf{ATT}_R) = (+\mathtt{r}_{x_1,y_1} \cup +\mathtt{r}_{y_1,x_1})\,; \ldots ; (+\mathtt{r}_{x_n,y_n} \cup +\mathtt{r}_{y_n,x_n})\,.$$

直观地说，对于每个 $(x,y) \in R$，$\mathsf{dis}(\mathsf{ATT}_R)$ 会使集合 $\{\mathtt{r}_{x,\ y}, \mathtt{r}_{y,\ x}\}$ 中至少一个变量为真。此外，对于对称关系$R^{\leftrightarrow}$中的每个 $(x,y) \in R^{\leftrightarrow}$，dis 会使 $\mathtt{r}_{x,\ y}$、$\mathtt{r}_{y,\ x}$ 中至少一个为真，也可能两者都为真。

现在，我们可以定义适用于 rIAF 的 makeComp 程序版本。给定 rIAF $=(A^F, R^F, A^?, R^?, R^{\leftrightarrow})$，令

$$\mathsf{makeComp}^{\mathsf{rIAF}} = \mathsf{mkTrueSome}(\mathsf{AW}_{A^?})\,; \mathsf{mkTrueSome}(\mathsf{ATT}_{R^?})\,; \mathsf{dis}(\mathsf{ATT}_{R^{\leftrightarrow}}).$$

下面的命题说明上述程序是正确的。

命题 4. 设 rIAF $=(A^F, R^F, A^?, R^?, R^{\leftrightarrow})$，则：

- 如果 $(v_{\mathsf{rIAF}}, v) \in ||\mathsf{makeComp}^{\mathsf{rIAF}}||$，那么 $(A_v, R_v) \in$ rIAF 的补全。
- 如果$(A^*,\ R^*) \in$补全(rIAF)，那么$(v_{\mathsf{rIAF}},\ v_{(A^*,R^*)}) \in ||\mathsf{makeComp}^{\mathsf{rIAF}}|| \quad |.$

[5]记住，$\mathsf{ATT}_R = \{\mathtt{r}_{x,y} \mid (x,y) \in R\}$，并且 ATT_R 是命题变量集合 $\mathsf{ATT}_{\mathcal{U}\times\mathcal{U}}$ 的子集。

再次强调，接受问题可以归约为 DL-PA模型检验问题。请注意，rIAF 的接受问题定义与 IAF 的定义相同（我们只需改变输入）。下面直接给出我们所需的归约结果：

命题 5. 令 $\sigma \in \{st, co, gr, pr, se, id, ea, na, stg\}$。令 rIAF $=(A^F, R^F, A^?, R^?, R^{\leftrightarrow})$，且 $a \in A^F$。则：

- 对于输入为 rIAF 和 a的 σ-NSA，答案为是，当且仅当$v_{\mathsf{rIAF}} \models [\mathsf{makeComp}_{\mathsf{rIAF}}; \mathsf{makeExt}^{\sigma}]$属于$a$。
- 对于输入为 rIAF 和 a 的 σ-NCA，答案为是，当且仅当$v_{\mathsf{rIAF}} \models [\mathsf{makeComp}_{\mathsf{rIAF}}]\langle \mathsf{makeExt}^{\sigma}\rangle$属于$a$。
- 对于输入为 rIAF 和 a的 σ-PCA，答案为是，当且仅当$v_{\mathsf{rIAF}} \models \langle \mathsf{makeComp}_{\mathsf{rIAF}}; \mathsf{makeExt}^{\sigma}\rangle$属于$a$。
- 对于输入为 rIAF 和 a 的 σ-PSA，答案为是，当且仅当$v_{\mathsf{rIAF}} \models \langle \mathsf{makeComp}_{\mathsf{rIAF}}\rangle [\mathsf{makeExt}^{\sigma}]$属于$a$。

4.3 缩小补全集合

不完整论证框架（及其扩展版本）以简单直观的方式处理论证情境中的不确定性。然而，我们能够用它们建模的情境类型相当有限（稍后将详细讨论）。这正是开发表达能力更强的形式体系的主要动机；事实上，去年出现了几种并行的提案，分别称为受约束的不完整论证框架 [48, 54]或带依赖关系的不完整论证框架[41, 42]。我们先介绍我们提出的受约束的不完整论证框架版本（即 [48]中引入的版本），然后再转向其他方法。

4.3.1 受约束的不完整论证框架

受约束的不完整论证框架（cIAF）是一个二元组 cIAF $=(A, \varphi)$，其中 $A \subseteq \mathcal{U}$ 是论证的集合，φ 是基于命题变量集合 $\mathsf{AW}_A \cup \mathsf{ATT}_{A\times A}$构造的布尔公式。[6] 给定 cIAF 的补全集合为

$$\mathsf{completions}(A, \varphi) = \{(A_v, R_v) \mid v \subseteq \mathtt{Prp}_A \text{ and } v \models \varphi\}.$$

例 3. 考虑 $\mathsf{cIAF}_0 = (A, \varphi)$，其中 $A = \{a, b\}$ 和 $\varphi = (\mathsf{aw}_a \wedge \mathsf{aw}_b) \wedge (\mathsf{r}_{a,b} \vee \mathsf{r}_{b,a}) \wedge \neg(\mathsf{r}_{a,b} \wedge \mathsf{r}_{b,a}) \wedge \neg \mathsf{r}_{a,a} \wedge \neg \mathsf{r}_{b,b}$。$\mathsf{cIAF}_0$的补全如下：



[6]我们对 cIAF 的原始定义做了少许修改 [48], ，将定义域从 $\mathcal{U}$ 改为任意的 A，因为这样可以自然地引入论证动态，正如我们将在第5.2节中所做的那样。

请注意，与之前讨论的结构类别不同，cIAF 的补全集合可能为空，因为 φ 可能是一个不一致公式。此外，即使 φ 是一致的，也可能没有任何表示非空论证框架的赋值使其成立，因此我们可能得到空论证框架（$\emptyset, \emptyset$）作为 cIAF 的唯一补全；例如，$\mathsf{completions}((\{a\}, \neg \mathsf{aw}_a)) = \{(\emptyset, \emptyset)\}$。

cIAF 的必要性。除了在数学上颇具趣味之外，人们或许还会疑惑为何要使用 cIAF。如前所述，我们的主要动机在于：尽管与此前介绍的形式体系（即 (r)IAF 及其子类）相关的推理任务的计算复杂度已得到充分研究且相对较低，但它们的建模能力却相当有限。举例来说，考虑一名主张方，她正在推测对手在一场非常简单的辩论中的观点，该辩论仅包含两个论证 $\{a, b\}$。假设 a 是右翼总统候选人提出的一个关于公共卫生政策的论证。同样，b 是左翼候选人提出的一个论证。设想 a 和 b 的结论相互矛盾，因此二者彼此不相容。这里，我们可以非正式地将 R 理解为击败关系，即一种基于逻辑不相容性以及对相关论证进行某种基于认知的评估（例如考察其前提的可靠性）的关系，这正是结构化论证中通常采用的做法。现在，假设我们的主张方知道她的对手具有两极化倾向，即对手已经倾向于政治光谱中的某一方，但她不知道究竟是哪一方；那么，该主体归因于其对手的可能论证框架，恰好就是 cIAF_0 的补全（见例 3）。稍后将证明（命题 8），不存在一个 rIAF（因而也不存在 IAF）能够具有与 cIAF_0 完全相同的补全集合。

现在说明如何在 DL-PA 中表示 cIAF。设 cIAF $= (A, \varphi)$，并将其关联赋值简单地定义为空集，即 $\mathsf{v}_{\mathsf{cIAF}} = \emptyset$。（实际上，对 Prp_A 上的任意赋值都可以完成此任务。）生成 cIAF 的所有补全的程序定义为

$$\mathsf{makeComp}^{\mathsf{cIAF}} = \mathsf{vary}(\mathsf{AW}_A); \mathsf{vary}(\mathsf{ATT}_{A\times A}); \varphi?.$$

$\mathsf{makeComp}^{\mathsf{cIAF}_0}$（见例 3）的行为如图 1 所示。

命题 6. 设 cIAF $= (A, \varphi)$，则：

- 若 $(\mathsf{v}_{\mathsf{cIAF}}, \mathsf{v}) \in ||\mathsf{makeComp}_{\mathsf{cIAF}}||$，则 $(A_v, R_v) \in$ cIAF 的补全。
- 若 $(A^*, R^*) \in$ cIAF 的补全，则 $(\mathsf{v}_{\mathsf{cIAF}}, \mathsf{v}_{(A^*,R^*)}) \in ||\mathsf{makeComp}_{\mathsf{cIAF}}||$。

针对 (r)IAF 的推理问题可以轻松地调整为针对 cIAF 的推理问题：我们只需确保提出查询所涉及的论证属于所有补全。例如，请看：

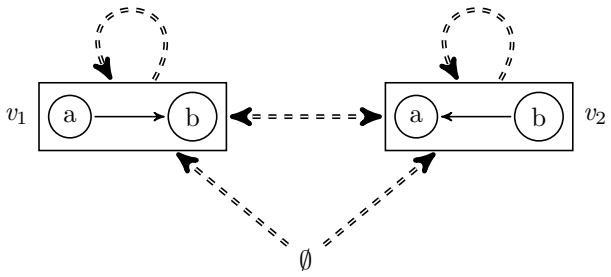


图 1：将 cIAF_0 的补全视为 $\mathsf{Prp}^{\{a,b\}}$ 上的赋值。虚线双箭头表示 $\mathsf{makeComp}^{\mathsf{cIAF}_0}$ 的解释；省略了 $\mathsf{Prp}^{\{a,b\}}$ 上的其他赋值。

σ-必要-可辩护接受（σ-NCA）
给定：一个受约束 IAF cIAF $= (A, \varphi)$ 以及一个论证 $a \in A$，使得 $\models \varphi \rightarrow \mathsf{aw}_a$。 问题：对于每一个，以下说法是否成立 $(A^*, R^*) \in$ 补全(cIAF) 是否存在一个 $E \in \sigma(A^*, R^*)$，使得 $a \in E$?

请注意，要求 $\models \varphi \rightarrow \mathsf{aw}_a$ 等同于要求 $a \in A$ 对于所有 $(A, R) \in$补全(A, φ)。

再一次，我们可以将 cIAF 中的接受问题归约为 DL-PA 模型检查问题。

命题 7. 设 cIAF $= (A, \varphi)$，并设 $a \in A$，使得 $\models \varphi \rightarrow \mathsf{aw}_a$。令 $\sigma \in \{st, co, gr, pr, se, id, ea, na, stg\}$。则：

- 输入为 cIAF 且 a 的 σ-NSA 的答案为是，当且仅当$v_{\mathsf{cIAF}} \models [\mathsf{makeComp}_{\mathsf{cIAF}}; \mathsf{makeExt}^{\sigma}]$属于$a$。
- 输入为 cIAF 且 a 的 σ-NCA 的答案为是，当且仅当$v_{\mathsf{cIAF}} \models [\mathsf{makeComp}_{\mathsf{cIAF}}]\langle \mathsf{makeExt}^{\sigma} \rangle$属于$a$。
- 输入为 cIAF 且 a 的 σ-PCA 的答案为是，当且仅当$v_{\mathsf{cIAF}} \models \langle \mathsf{makeComp}_{\mathsf{cIAF}}; \mathsf{makeExt}^{\sigma} \rangle$属于$a$。
- 对于输入为 cIAF 和 a 的 σ-PSA，答案为是，当且仅当$v_{\mathsf{cIAF}} \models \langle \mathsf{makeComp}_{\mathsf{cIAF}} \rangle [\mathsf{makeExt}^{\sigma}]$属于$a$。

我们注意到，除了这些推理问题之外，还可以考虑检查 cIAF 的补全集合是否为空这一推理任务。

4.3.2 密切相关的方法

如前所述，缩小不完全论证框架的补全集这一想法，在过去一年中催生了多个并行的提案。本小节将简要介绍与我们针对 [48] 提出的 cIAF 不同的两种方法。

cIAF 的一种更偏图论的版本。在 [54], Jean-Guy Mailly 定义了他所提出的 cIAF 版本，为避免混淆，我们在此将其称为 cIAF^{JM} 。cIAF^{JM} 是形如 (IAF φ) 的一对，其中 IAF $= (A^F, R^F, A^?, R^?)$ 是一个不完全论证框架，且φ 是关于 $\mathsf{Prp}^{\mathsf{IAF}} = \mathsf{AW}_{A\cup A^?} \cup \mathsf{ATT}_{(A\cup A^?)\times(A\cup A^?)}$ 的布尔公式。于是，cIAF^{JM} = (IAF φ) 的补全集定义为

$$\mathsf{completions}(\mathsf{IAF}) \cap \{(A_v, R_v) \mid v \subseteq \mathtt{Prp}^{\mathsf{IAF}} \text{ and } v \models \varphi\}.$$

带依赖关系的IAF。在 [41, 42], 中，由 Bettina Fazzinga、Sergio Flesca 和 Filippo Furfaro 组成的卡拉布里亚大学团队提出了带依赖关系的IAF这一概念。[7] 更确切地说，他们提出的两种方案分别聚焦于我们已经提到过的两类受限IAF：arg-IAF 和 att-IAF。为简洁起见，这里仅介绍 [41]提出的带依赖关系的 arg-IAF 概念。令 A 为论证集合，并令 X、$Y \subseteq A$。首先， A 上的依赖关系是指 $X \Rightarrow Y$ 或 OP(X)，其中 OP $\in$ {OR、NAND、CHOICE}。其次，带依赖关系的 arg-IAF（简称 d-arg-IAF）是一个三元组 $((A, A^?, R)\ \Delta)$，其中 $(A, A^?, R)$ 是一个 arg-IAF，而 Δ 是 $A^?$上的依赖关系集合。在定义 d-arg-IAF 的补全之前，我们需要明确如何解释 arg-IAF 中的依赖关系。令 (A, R) 为一个论证框架，并令 δ 为 A 上的依赖关系。我们称 (A, R) 满足 δ[8] ，当且仅当以下互斥条件之一成立：

- $\delta = X \Rightarrow Y$且（如果 $X \subseteq A$，那么 $A \cap Y \neq \emptyset$）；
- $\delta =$ 或(X) 且 $A \cap X \neq \emptyset$,
- $\delta =$ 与非(X) 且 $A \cap X \subset X$,
- $\delta =$ 选择(X) 且 $|A \cap X| = 1$。

$((A, A^?, R)\ \Delta)$ 的补全定义为满足每个依赖 $\delta \in \Delta$ 的 arg-IAF $(A, A^?, R)$ 的补全。

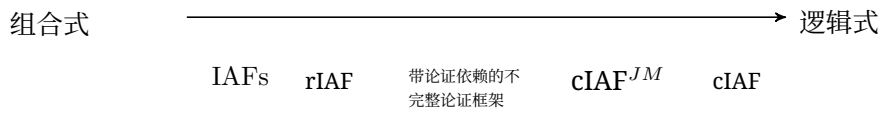
这三种备选方案已在 [52] 中进行了比较。下一节将进一步提出一些新的见解。这里先指出两点。首先，请注意，cIAF 的两个版本以及带依赖关系的 IAF 显然都受到受约束抽象论证框架这一概念的启发 [29], ；受约束抽象论证框架是形如 $((A, R)\ \varphi)$ 的二元组，其中 φ 用于缩小 (A, R) 的扩展集合。其次，请注意，与这两类结构相关的推理任务显然都可以编码为 DL-PA 中的形式，但这里不展开细节。这里只需指出，每个依赖关系集合 Δ 都可以转换为布尔公式 $t(\Delta)$，然后在 $\mathbf{v}^{(A,R)}$ 处执行程序 $\mathsf{makeComp}_{((A,A^?,R)\ \Delta)} = \mathsf{vary}(\mathsf{AW}_{A^?}); t(\Delta)?$，即可计算出 $((A, A^?, R)\ \Delta)$ 的所有补全。

[7]术语 *correlations* 在 [41, 42] 中用作 *dependencies* 的非正式对应说法。为避免混淆，我们采用后一个术语。[8][41] 使用了表达式“(A, R) 相对于 δ 是有效的”，但在逻辑分析中，我们的表达方式更为恰当。

4.4 不同方法的比较

现在来比较表示关于论证框架的定性不确定性的不同方法。我们先作几点一般性说明。

组合方法与逻辑方法。关于不完全论证框架的开创性研究，其核心思想是将补全定义为有向图，其论域和关系介于给定的区间之间，以此表示不确定性。这种方法可以称为“组合式”，因为一旦给出区间的两个极端（例如 A^F 和 $A^F \cup A^?$），计算补全就等同于找出区间内所有可能的组合。随着研究的推进，我们或许可以称为“逻辑式”的其他推理特征也逐渐被纳入补全的定义中。例如，rIAF 通过加入 $R^{\leftrightarrow}$ 引入了某种析取推理。我们可以将从组合方法到逻辑方法的转变理解为一种光谱：



请注意，在最右端（cIAF）处，补全的组合式特性已完全消失。

图形表示。IAF 的一个吸引人之处在于，它们支持非常直观的图形表示（见例 1）。有趣的是，rIAF 和 d-arg-IAF 也可以完全以图示方式表示；有关 d-arg-IAF 的示例，请见[49] 。正如 [53], 所指出的，cIAF^{JM} 只能支持部分图形表示。最后，我们的 cIAF 丧失了这种图示可表示性，因为它们完全抽象掉了 IAF 的图论定义。因此，在这方面，带依赖关系的 IAF 相较于 cIAF 和 cIAF^{JM}更具优势。

通过补全集表达力。根据 [53], ，我们可以使用各形式体系能够表示的补全集，比较前述各形式体系在不确定性论证方面的建模能力。令 $\mathcal{IAF}$ 表示所有 IAF 的类，对于 att-$\mathcal{IAF}$、 arg-$\mathcal{IAF}$、 $\mathcal{RIAF}$、 d-arg-$\mathcal{IAF}$、 d-att-$\mathcal{IAF}$、c-$\mathcal{IAF}$和 c-$\mathcal{IAF}^{JM}$亦同。令 $\mathcal{X}$和 $\mathcal{Y}$为表示这些类之一的元变量。若对于每个 $Y \in \mathcal{Y}$，都存在一个 $X \in \mathcal{X}$，使得 completions(X) $=$ completions(Y)，则称 $\mathcal{X}$ 至少与 $\mathcal{Y}$ 具有同等表达力（记作： $\mathcal{X} \succeq \mathcal{Y}$）。我们用 $\succ$ 表示 $\succeq$的严格部分，用 $\preceq$ 表示 $\succeq$的逆关系，并用 $\equiv$ 简写 $\succeq \cap \preceq$。例如， [53] 中证明了 $\mathcal{RIAF} \succ \mathcal{IAF}$。

命题 8. cIAF 的表达能力严格强于 IAF 和 rIAF。换言之，对于每个 (r)IAF，都存在一个具有相同补全集合的 cIAF；但也存在一个 cIAF，使得没有任何 (r)IAF 具有相同的补全集合。

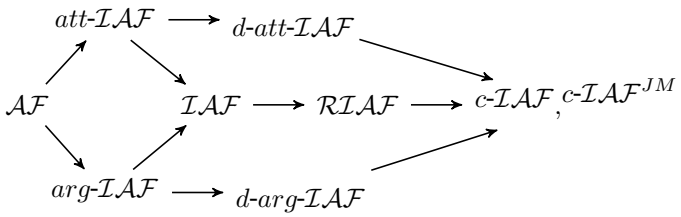


图2：形式论证中定性不确定性形式体系的相对表达力。从 $\mathcal{X}$ 到 $\mathcal{Y}$ 的箭头表示 $\mathcal{X} \preceq \mathcal{Y}$，即 $\mathcal{Y}$ 的表达力至少与 $\mathcal{X}$ 一样强。省略了自反和传递箭头。

在前一个命题证明的第一部分中——见附录——我们使用了一种论证，该论证适用于定义域为$\mathcal{U}$的任意有向图集合（而不仅仅适用于某个给定 rIAF 的补全），因此我们可以得出：

推论1. 对于定义域为 $\mathcal{U}$ 的任意有向图集合 S，都存在一个 cIAF cIAF，使得 S $=$ 补全(cIAF)。

换言之，cIAF 是一种表达力最强的形式体系，可用于表示关于论证框架的定性不确定性。采用与命题 8 证明中类似的论证，我们可以得出以下一般性结果：

命题9. 图2中的关系成立，其中从 $\mathcal{X}$ 到 $\mathcal{Y}$ 的箭头表示 $\mathcal{X} \preceq \mathcal{Y}$，且省略了传递和自反箭头。

除了提供完整的表达能力图外，该命题还凸显了一个事实：带依赖关系的 IAF 尚未形成表达力最强的表述。也就是说，我们有带依赖关系的 arg-IAF [41]，和带依赖关系的攻击不完整论证框架 [42]，，但没有带依赖关系的 IAF。因此，这些结构尚不能表达任意一组补全（与 cIAF 和 cIAF^{JM} 都能做到的情况相反）。显然，这些形式体系的混合版本也能达到表达力最大化。不过，仍有一些重要的设计选择需要作出；例如，是否允许混合依赖关系（涉及不确定论证和攻击的依赖关系）。

5 完全扩展动态与不确定性

正如引言中所述，论证框架（AF）遗漏了论证的两个基本方面：关于相关论证信息的不确定性（即在辩论过程中应考虑哪些论证和攻击），以及此类信息的动态变化。上一节中，我们讨论了表示论证框架不确定性的多种方式。至于论证框架的动态变化，这如今已是一个经过充分研究的领域；近期综述请参见 [38, 11]。本节概述如何将这两种思想结合起来。我们首先介绍一个经过充分研究的案例：控制论证框架 [31]，，并展示其主要推理任务同样可以用 DL-PA 编码。在指出其部分

局限性后，我们将继续研究一种扩展，它结合了 CAF 所涵盖的动态变化类型与 cIAF 在表示不确定性方面的灵活性。本节最后概述论证框架动态变化与不确定性的一般理论，为今后的研究提供概念工具。

5.1 控制论证框架

控制论证框架在 [31] 中被提出，并在 [32] 中应用于基于论证的协商。它们以一种联合方法处理论证框架的不确定性与动态性。在不确定性方面，它们的表达能力与 rIAF 相同（第 4.2 节）。在动态性方面，它们在每个补全层面刻画了所谓正规扩展的一种参数化版本 [9] 。

形式上，控制论证框架是一个三元组 CAF $=(F,U,C)$，其中：

- $F=(A^F,R^F)$ 是固定部分，其中 $R^F\subseteq(A^F\cup A^?)\times(A^F\cup A^?)$，并且 A^F 和 $A^?$ 均为论证的有限集；
- $U=(A^?,(R^?\cup R^{\leftrightarrow}))$ 是不确定部分，其中

$$R^?,R^{\leftrightarrow}\subseteq(A^F\cup A^?)\times(A^F\cup A^?)$$

且 $R^{\leftrightarrow}$ 具有对称性和反自反性；[9]

- $C=(A^C,R^C)$ 是控制部分，其中 A^C 是另一个有限论证集，并且

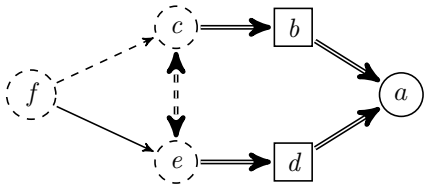
$$R^C\subseteq(A^C\times(A^F\cup A^?\cup A^C))\cup((A^F\cup A^?\cup A^C)\times A^C);$$

- A^F、$A^?$和 A^C 两两不相交；并且
- R^F、$R^?$ $R^{\leftrightarrow}$和 R^C 两两不相交。

我们将 $\mathcal{CAF}$ 记作所有控制论证框架的类。

给定一个 CAF $=(F,U,C)$，控制配置是控制论证 $CFG\subseteq A^C$的一个子集。直观地说，每个控制配置都可以看作主张方可能采取的一种论证行动。与 CFG相关联的 CAF 是 $\text{CAF}_{CFG}=(F,C_{CFG},U)$，其中 $C_{CFG}=(CFG,R^C\restriction_{A^F\cup A^?\cup CFG})$。

例4。考虑 CAF $\text{CAF}_0=(F_0,C_0,U_0)$，其中 $A_0^F=\{a\}$,$R_0^F=\{(f,e)\}$, $A_0^?=\{c,e,f\}$, $R_0^?=\{(f,c)\}$, $R_0^{\leftrightarrow}=\{(c,e),(e,c)\}$, $A_0^C=\{b,d\}$，以及 $R_0^C=\{(b,a),(d,a),(c,b),(e,d)\}$。我们将 CAF_0 图形化表示如下：



[9] $R^{\leftrightarrow}$ 的对称性和非自反性并非原始论文 [31], 中的假设，而是后来在关于控制论证框架的文献 [58, 56]中出现的。请注意，这两个假设都不会影响控制论证框架的表达力（按照第 4.4 节中的定义）。

对于控制论证框架，补全的概念定义如下：

- $(A^F \cup A^C) \subseteq A^* \subseteq (A^F \cup A^C \cup A^?)$；
- $(R^F \cup R^C) \restriction_{A^*} \subseteq R^* \subseteq (R^F \cup R^C \cup R^? \cup R^{\leftrightarrow}) \restriction_{A^*}$；以及
- 对于每个 $x, y \in A^*$: $(x, y) \in R^{\leftrightarrow}$ 蕴含 $(x, y) \in R^*$ 或 $(y, x) \in R^*$。

根据这一定义，控制论证/攻击一旦传达出去，其作用就如同固定论据/攻击。因此，CAF_0 的补全与 rIAF_0（例2）的补全一致，即表2中所示的补全。

关于控制论证框架，定义相关推理任务会稍微复杂一些，因为我们必须考虑其动态维度。在此语境下，一个自然的推理任务是找出一个控制配置（即一个控制论证集合），使得对手在得知这些论证后接受某个特定论证。和之前一样，可接受性取决于对补全和扩展的量化。下面举例说明：

σ-必要-怀疑式-可控性（σ-NSCon）
给定：一个控制论证框架 $\mathsf{CAF} = (F, U, C)$ 和一个论证 $a \in A^F$。 问题：是否存在一个配置 $CFG \subseteq A_C$，使得对于每个补全（A^*, R^*） 关于 CAF_{CFG}，以及每个 $E \in \sigma(A^*, R^*)$ $a \in E$?

现在我们来解释如何在 DL-PA 中对控制论证框架进行推理。就不确定性而言，控制论证框架本质上是内容丰富的不完整论证框架，因此编码过程中的难点在于其动态组件，即控制部分。

首先，给定一个 CAF $\mathsf{CAF} = (F, U, C)$，我们将其关联赋值定义为

$$\begin{aligned} v_{\mathsf{CAF}} &= v_{(A^F, R^F \cup R^C)} \\ &= \mathsf{AW}_{A^F} \cup \mathsf{ATT}_{R^F} \cup \mathsf{ATT}_{R^C} \\ &= \{\mathtt{aw}_x \mid x \in A^F\} \cup \{\mathtt{r}_{x,y} \mid (x, y) \in R^F\} \cup \{\mathtt{r}_{x,y} \mid (x, y) \in R^C\}. \end{aligned}$$

请注意，v_{CAF} 包含所有对应于控制攻击的攻击变量，但它们均未出现在$(A_{v_{\mathsf{CAF}}}, R_{v_{\mathsf{CAF}}})$中，因为尚未传达任何控制论证。这表明，在对控制论证框架进行认知解释时，主张方知道对手将如何看待涉及所有可传达论证的攻击关系。

为刻画 CAF 的动态组件，我们定义如下程序：

$$\mathsf{control}^{\mathsf{CAF}} = \mathsf{mkTrueSome}(\mathsf{AW}_{A^C}).$$

直观地说，控制^{CAF} 以非确定方式选择 CAF 的某些可能控制配置，即控制论证的某个子集。

一旦我们计算出某个控制配置，就会对 rIAF 使用相同的程序来计算补全：

$$\mathsf{makeComp}^{\mathsf{CAF}} = \mathsf{mkTrueSome}(\mathsf{AW}_{A^?}); \mathsf{mkTrueSome}(\mathsf{ATT}_{R^?}); \mathsf{dis}(\mathsf{ATT}_{R^{\leftrightarrow}}).$$

我们再次给出一个正确性结果：

命题10。设 CAF $=(F,U,C)$。

- 如果 $(\mathsf{v}_{\mathsf{CAF}}, \mathsf{v}) \in ||$控制$_{\mathsf{CAF}}$; makeComp$_{\mathsf{CAF}}||$，那么存在一个控制配置 $CFG \subseteq A^C$ 以及 CAF_{CFG} 的一个补全 (A^*, R^*)，使得 $(A_v, R_v) = (A^*, R^*)$。

- 对于每个控制配置 $CFG \subseteq A^C$ 以及每个 $(A^*, R^*) \in$ CAF，$\in$ 的补全，都存在一个赋值 $\mathsf{v} \in 2^{\mathsf{Prp}}$，使得 $(\mathsf{v}_{\mathsf{CAF}}, \mathsf{v})$ control$_{\mathsf{CAF}}$; makeComp$_{\mathsf{CAF}}$且 (,) (,)。

然后，我们可以将前述程序与 makeExt 结合起来，从而将可控性问题归约为 DL-PA 模型检查问题。

命题 11。设 $\sigma \in \{st, co, gr, pr, se, id, ea, na, stg\}$。设 CAF $=(F,U,C)$ 且 $a \in A^F$。那么：

- 对于输入为 CAF 且包含 a 的 σ-NSCon，答案为是，当且仅当$v_{\mathsf{CAF}} \models \langle$控制$_{\mathsf{CAF}}\rangle$[makeComp$_{\mathsf{CAF}}$;makeExt$^{\sigma}$]属于$a$。
- 对于输入为 CAF 且包含 a 的 σ-NCCon，答案为是，当且仅当$v_{\mathsf{CAF}} \models \langle$控制$_{\mathsf{CAF}}\rangle$[makeComp$_{\mathsf{CAF}}$]$\langle$makeExt$^{\sigma}\rangle$属于$a$。
- 对于输入为 CAF 且包含 a 的 σ-PCCon，答案为是，当且仅当$v_{\mathsf{CAF}} \models \langle$控制$_{\mathsf{CAF}}$; makeComp$_{\mathsf{CAF}}$; makeExt$^{\sigma}\rangle$属于$a$。
- 对于输入为 CAF 且包含 a 的 σ-PSCon，答案为是，当且仅当$v_{\mathsf{CAF}} \models \langle$控制$_{\mathsf{CAF}}$; makeComp$_{\mathsf{CAF}}\rangle$[makeExt$^{\sigma}$]属于$a$。

我们通过强调并明确指出 CAFs 的两个主要建模局限，来结束本节；这些局限我们此前已经提及。就不确定性而言，它们无法超越 rIAF。就动态而言，它们所建模的通信形式假定不确定性不会增加。更正式地说：

注释1。 $\mathcal{CAF} \equiv \mathcal{RIAF}$。

注释2。设 CAF $=(F,U,C)$，且 $CFG, CFG' \subseteq A^C$。那么| completions(CAF$_{CFG}$)| = |completions(CAF$_{CFG'}$)|。

5.2 受控约束不完整论证框架

迄今介绍的各种方法有一个优点：它们可以自由组合。此外，DL-PA 中这些形式体系的编码很容易推广到组合结构类。本小节将 cIAF 所建模的不确定性与 CAF 所建模的动态结合起来，以展示这种灵活性。形式上，受控约束不完整论证框架（CcIAF）是一个元组 CcIAF $=(A^C, R^C, A^S, \varphi)$，其中：

- A^C （控制论证）和 A^S （静态论证）互不相交，
- $R^C \subseteq (A^C \times (A^C \cup A^S)) \cup ((A^C \cup A^S) \times A^C)$，
- φ 是关于 $\mathsf{AW}_{A^S} \cup \mathsf{ATT}_{A^S \times A^S}$ 的布尔公式。

给定 CcIAF $=(A^C, R^C, A^S, \varphi)$，有序对 (A^S, φ) 是其底层 cIAF。

通过结合 CAF 和 cIAF 背后的直观含义，将补全的概念应用于 CcIAF；也就是说，(A^C, R^C, A^S, φ) 的一个补全是任意满足以下条件的论证框架 (A^*, R^*)：

- $A^* = A^C \cup A'$,
- $R^* = (R^C \cup R') \restriction_{A^*}$,
- (A', R') 是底层 cIAF 的一个补全。

控制配置的概念也可以直接推广到我们这一类新的结构中。具体而言，CcIAF $=(A^C, R^C, A^S, \varphi)$ 的一个控制配置是任意一个 $CFG \subseteq A^C$。与 CFG 相关联的 CcIAF 定义为 $\mathsf{CcIAF}_{CFG} = (CFG, R^C \restriction_{CFG}, A^S, \varphi)$。我们可以将可控性问题推广到 CcIAF：

σ-必要-怀疑式-可控性（σ-NSCon）
给定：一个控制受限的不完全论证框架 CcIAF $=(A^C, R^C, A^S, \varphi)$ 和一个论证 $a \in A^S$ s.th. $\models \varphi \to \mathsf{aw}_a$。 问题：是否存在一个配置 $CFG \subseteq A_C$，使得对于每个补全 (A^*, R^*) 对于 CcIAF_{CFG} 的每个 $E \in \sigma(A^*, R^*)$ $a \in E$，这一说法是否成立?

关于我们刚刚定义的推理问题的 DL-PA 编码，我们首先为每个 CcIAF 指定其对应的赋值，方法与我们处理 CAF 和 cIAF 时类似：

$$\begin{aligned} v_{\mathsf{CcIAF}} &= \mathsf{ATT}_{R^C} \\ &= \{\mathtt{r}_{x,y} \mid (x,y) \in R^C\}. \end{aligned}$$

CcIAF 的控制部分采用与 CAFs 中相同的 DL-PA 程序进行编码：

$$\mathsf{control}^{\mathsf{CcIAF}} = \mathsf{mkTrueSome}(\mathsf{AW}_{A^C}).$$

计算补全的程序也有类似情况，我们沿用了用于 cIAF 的程序：

$$\mathsf{makeComp}^{\mathsf{CcIAF}} = \mathsf{vary}(\mathsf{AW}_{A^S}); \mathsf{vary}(\mathsf{ATT}_{A^S \times A^S}); \varphi?.$$

命题 12. 令 CcIAF $= (A^C, R^C, A^S, \varphi)$。

- 若 $(\mathrm{v}_{\mathsf{CcIAF}}, \mathrm{v}) \in |\text{控制}_{\mathsf{CcIAF}}; \mathsf{makeComp}_{\mathsf{CcIAF}}|$，则存在一个控制配置 $CFG \subseteq A^C$ 以及 CcIAF_{CFG} 的一个补全 (A^*, R^*)，使得 $(A_v, R_v) = (A^*, R^*)$。

- 对于每个控制配置 $CFG \subseteq A^C$ 以及每个 $(A^*, R^*) \in \mathsf{completions}(\mathsf{CcIAF}_{CFG})$，存在一个赋值 $\mathrm{v} \in 2^{\mathsf{Prp}_{A^S \cup A^C}}$，使得 $(\mathrm{v}_{\mathsf{CAF}}, \mathrm{v}) \in |\text{控制}_{\mathsf{CcIAF}}; \mathsf{makeComp}_{\mathsf{CcIAF}}|$ 且 $(A_v, R_v) = (A^*, R^*)$。

再一次，我们也可以将涉及 CcIAF 的推理任务归约为 DL-PA 模型检测问题。细节留给读者。

5.3 迈向一般理论

在 [38], 中，研究者构建了一个关于抽象论证系统动态的一般理论。本文的重点是论证框架的动态，但正如作者所指出的，该理论乍看之下也适用于其他类型的论证框架。在本小节中，我们将他们的分类应用于第4节所研究的、用于表示论证框架定性不确定性的形式体系。同时，我们展示了 DL-PA 如何作为形式化这一一般理论诸多部分的良好逻辑候选。

结构约束。根据 [38], ，在论证系统中，人们可能希望强制施加不同类型的约束。第一类约束涉及论证框架的结构。 [38] 区分了基本结构约束和全局结构约束。前者直接定义在框架的组成部分上（添加/移除某些论证/攻击）；后者则要求输出的论证框架满足某种性质，例如无奇数环、无环等。两类约束对于我们在第4节研究的结构类型都完全合理。有趣的是，这些形式体系更丰富的性质还允许作出进一步区分。

基本结构约束。在论证框架中，基本约束相当于添加/移除论证/攻击（或这些操作的组合，如抽象论证框架扩展 [9] 的情况），而在具有定性不确定性的论证框架中，我们可以执行更微妙的操作。下面以不完全论证框架为例，说明其中一些操作。

确定不确定论证／攻击。在辩论中，主体可能希望通过“确定”某个不确定论证／攻击来提升其认知状态。形式上，且将讨论范围限定为论证和不完全论证框架，

给定 IAF $= (A^F, R^F, A^?, R^?)$ 和 $a \in A^?$，定义偏函数 settle : $(\mathcal{IAF} \times \mathcal{U}) \to \mathcal{IAF}$ 如下：

$$\mathsf{settle}(\mathsf{IAF}, a) = (A^F \cup \{a\}, A \setminus \{a\}, R^F, R^?).$$

在 DL-PA 中，我们可以直接计算所得 IAF 的补全：

$$\mathsf{completions}(\mathsf{settle}(\mathsf{IAF}, a)) = \{(A_v, R_v) \mid (v_{\mathsf{IAF}}, v) \in ||\mathsf{makeComp}^{\mathsf{IAF}}; +\mathtt{aw}_a||\}.$$

变得不确定的通信论据。另一种动态可通过将论据从 $\mathcal{U} \setminus (A^F \cup A^?)$ 移至 $A^?$ 来形式化建模，可用于描述这样的情形：论证通过一个并非完全可信的通信渠道（例如即时通讯应用）进行，因此主张方无法确定对手是否收到了发送的论据。同样，所得 IAF 的补全可以在 DL-PA 中轻松计算；同样的思路也可用于通信攻击，而非通信论据。

效果不确定的通信论据。还有另一种行动，即通信论据，而这些论据对对手框架产生的影响并不明确。例如，在 CAF 的背景下，可以通过扩展定义域和范围的定义来放宽其定义，使其包含 $R^?$ 或 $R^{\leftrightarrow}$，从而将 A^C 纳入其中。

逻辑结构的信念改变方法。对于 cIAF（这也适用于 cIAF^{JM}），基本变化包括扩展或缩小定义域 A，或者更有趣的是，改变认知约束 φ。对于后一种情况，可以采用信念改变文献中的方法；例如，如果要添加一条与 φ 不一致的新信息 ψ，可以通过 AGM 信念修正算子 [1]来实现。在这方面，DL-PA 已被证明可用于刻画信念改变算子 [47], ，而这些算子随后也已应用于论证框架 [35, 36]。

基本结构变化的类型。我们可以运用若干有趣的标准，对具有定性不确定性的论证框架中的基本结构变化进行分类。这里仅指出其中两个。就对论据的认知而言，我们可以区分内部行动、获得论据的行动和失去论据的行动。简单来说，内部行动的结果是，主体既不会意识到任何新论据，也不会对任何新论据失去认知。[10] 更形式化地说，并且将讨论范围限定在不完全论证框架（IAFs）内，我们称将 $\mathsf{IAF}_0 = (A_0^F, A_0^?, R_0^F, R_0^?)$ 转变为 $\mathsf{IAF}_1 = (A_1^F, A_1^?, R_1^F, R_1^?)$ 的行动为内部行动，当且仅当 $A_0^F \cup A_0^? = A_1^F \cup A_1^?$。例如，上文定义的部分函数确定函数 settle : $(\mathcal{IAF} \times \mathcal{U}) \to \mathcal{IAF}$ 显然是内部行动。

[10]然而，它们可能会改变其所知晓的论证的认知状态。

形式上，获得论据的行动要求 $A_0^F \cup A_0^? \subset A_1^F \cup A_1^?$，使主体至少意识到一个新论据。我们上文说明的传递不确定论据这一行动，就是获得论据的行动的一个例子。最后，对于失去论据的行动，有 $A_1^F \cup A_1^? \subset A_0^F \cup A_0^?$，也就是说，主体至少对一个论据失去了认知。[11] 对基本结构变化进行分类的第二个标准，是衡量它们对补全数量的影响，因为直观而言，补全越多，形式化代理所面对的不确定性就越大。例如，上文描述的确定函数总会减少补全的数量；计算控制论证框架的控制配置会使补全数量保持不变（精确表述见注释2）；而传递不确定论据（同样如上文所述）则会增加补全的数量。

全局结构约束。在论证框架中，全局结构约束涉及获得无环图、无奇环图等。这些约束的动机在于它们所带来的良好数学性质。例如，自 [39] 以来人们已知，在无环论证框架中，四种经典语义会合并为同一种语义。有趣的是，DL-PA 可以表达其中许多约束。例如，在 [36] 中，构造了刻画奇数长度环和偶数长度环存在性的多项式公式。将这些约束推广到本文研究的更复杂形式体系时，全局约束可以被要求可能成立（即在至少一个补全中成立），也可以被要求必然成立（即在所有补全中成立）。此外，还可以使用 DL-PA 检查该约束是否可能成立或必然成立。更准确地说，为简单起见，聚焦于不完全论证框架：设 φ 是刻画某一目标全局约束的公式，并设 IAF 是一个不完全论证框架；则有，φ 可能成立（分别地，必然成立）当且仅当 $\mathbf{v}_{\mathsf{IAF}} \models \langle \mathsf{makeComp}^{\mathsf{IAF}} \rangle \varphi$（分别地，当且仅当 $\mathbf{v}_{\mathsf{IAF}} \models [\mathsf{makeComp}^{\mathsf{IAF}}]\varphi$）。在论证框架中，全局约束通常通过基本变化（即上述变化）来强制实现。同样，这种关系也可以在 DL-PA 中进行研究。例如，如果我们想知道某个全局约束 φ 是否会由于解决 $a \in A^?$而在 IAF 中可能得到强制实现，那么只需对 $\mathbf{v}_{\mathsf{IAF}} \models \langle \mathsf{makeComp}^{\mathsf{IAF}}; +\mathsf{aw}_a \rangle \varphi$ 是否成立进行模型检验即可。

可接受性约束。[38] 区分的第二类约束关注论证系统中论证评估过程的输出。这里同样适用基本/全局之分。若限定于论证框架，人们可能希望强制一组论证成为至少一个（或每个）扩展的一部分（或与之相等）；这就是基本可接受性约束。自 [9], 以来，这类强制大概是抽象论证文献中研究最多的，因为它

[11] 最后一种行动与近期一项在论证框架语境下对 *forgetting an argument* [12] 这一概念的深入研究相关。有趣的是，我们可以在 IAF 中刻画 [33], 所作的区分，即 *forgetting-as-becoming-unaware*（将一个论证从$A^F \cup A^?$ 移至 $\mathcal{U} \setminus (A^F \cup A^?)$）与 *forgetting-as-becoming-ignorant*（将一个论证从A^F 移至 $A^?$）之间的区别。

在现实生活中的论证中有一个清晰的非正式对应：说服对手，基本上就是强制某些目标论证。此外，从更技术性的角度来看，人们也可能希望强制某种全局可接受性约束：控制扩展集合的基数、其结构等。同样，定性不确定性引入了一个新的量化层面：可接受性强制可以是可能的（即至少在一个补全中实现），也可以是必然的（即在所有补全中实现）。与论证框架的情况一样，可接受性约束通常通过我们上文所述的结构变化（或其组合）来强制实现。例如，控制论证框架和 CcIAF 的推理任务，就是通过执行一系列不会增加不确定性的基本结构变化（激活控制论证），来强制实现可能或必然的可接受性约束。

语义约束。最后，[38] 所区分的第三类约束会影响用于评估论证的语义。简而言之，施加语义约束就意味着改变论证评估过程中所采用的标准。就此而言，不仅可以将参数 σ 切换为 σ'，还可以从宽容可接受性转向怀疑可接受性，反之亦然。与之前一样，第 4 节研究的形式体系引入了额外一层量化，在表述语义约束时必须予以考虑：我们可以从“可能”语义（论证应当在至少一个补全中被接受）转向“必然”语义（论证应当在所有补全中都被接受），反之亦然。正如本文所展示的（例如，命题 3），可能可接受性与必然可接受性之间的区别可以通过 DL-PA 清晰地表达。

6 讨论、相关工作与未来方向

我们将论证框架及其扩展的逻辑编码又向前推进了一步，从命题逻辑和量化布尔公式（QBF）中的编码，转向简单版本的动态逻辑 DL-PA 中的编码。基于 SAT 求解器的论证推理问题方法，通常采用 Besnard 和 Doutre 对论证框架及其语义进行的命题逻辑编码 [19], ，并将其扩展到 QBF，以处理需要进行极大性检验的语义；近期的此类方法可参见[57] ，抽象论证推理器方法综述可参见 [26] 。基于我们的工作，可以使用 DL-PA 模型检查器取代 SAT 求解器，自动判定我们在此研究的推理问题。不过，这还得等待此类模型检查器的出现，目前它们尚不存在。另一种做法是将 DL-PA 转换为 QBF，并使用针对后者的求解器。IRIT 的 LILaC 研究组目前正在推进这一方向。

总体而言，我们在 DL-PA 中完成的所有工作，同样也可以在表达能力相当的

逻辑框架中完成，例如命题逻辑或量化布尔公式。与前者相比，DL-PA 的优势在于：(1) 某些语义在 DL-PA 中可以表达得更紧凑，例如优选语义：它在多项式层次中比其他语义高一个层级，因此无法用多项式规模的命题逻辑公式刻画；而在 [36],[12] 中给出了一个多项式规模的 DL-PA 公式；(2) 推理问题可以直接表示为 DL-PA 程序。与量化布尔公式相比，DL-PA 的优势在于，通过程序对推理问题进行编码，比文献中常见的复杂量化布尔公式编码更加自然。事实上，大多数关于定性不确定性下论证的研究，都采用量化布尔公式编码和算法来确定相关推理任务的复杂度（例如参见 [13] 或 [58]）。[36] 已指出的所有使用 DL-PA 而非量化布尔公式来编码论证语义的优势，在我们的编码中都得以保留。尤其是，“诸如 makeExt$^{\sigma}$ 这样的扩展构造程序，能够以比量化布尔公式编码更加通用、灵活且自然的方式刻画事物”。

朝着定理证明方法更进一步。我们对带有定性不确定性的论证形式系统所作的编码可以称为混合方法，因为它结合了以往的语义推理与 DL-PA 内部的推理。例如，为了计算一个 IAF 的补全，首先需要找到其关联的赋值（这一步是在逻辑之外、利用语义对象进行推理），然后写出 makeComp 程序，最后在 DL-PA 中进行推理，以找出该关联赋值的 makeComp 后继。我们采用这种混合方法，是因为我们认为，将有向图与 Prp 上的命题赋值对应起来十分直观。不过，我们可以采用 [35, 37, 36] 中的结果，在这里获得一种更加统一的方法。例如，给定 IAF $=(F,U)$，我们不必计算其关联赋值，而是可以写出一个刻画其不动元素的命题公式（类似于 [35] 中对标准 AF 所做的工作，以及我们在附录中命题 8 的证明）：

$$\mathsf{Th}(\mathsf{IAF}) = \bigwedge_{x\in A^F} \mathtt{aw}_x \wedge \bigwedge_{x\in\mathcal{U}\setminus A^F} \neg\mathtt{aw}_x \wedge \bigwedge_{(x,y)\in R^F} \mathtt{r}_{x,y} \bigwedge_{(x,y)\in\mathcal{U}\times\mathcal{U}\setminus R^F} \neg\mathtt{r}_{x,y}.$$

如果将该公式与 makeComp 程序和逆算子结合起来，我们将得到一个公式，其模型完全刻画了 IAF 的补全集合：

$$\mathsf{completions}(\mathsf{IAF}) = \{(A_v, R_v) \mid v \in ||\langle\big(\mathsf{Th}(\mathsf{IAF})?; \mathsf{makeComp}^{\mathsf{IAF}}\big)^{\smile}\rangle\top||\}.$$

相较于我们的会议论文 [48]，本文有以下新内容。本文基于我们此前的会议论文 [48], ，并沿三个主要方向对其进行了改进和扩展。首先，在第 3 节中，(i) 我们刻画了论证

[12]请注意，我们对 [36] 中公式 Preferred 的改编，在更一般的背景论证集合 $\mathcal{U}$ 的设置下捕捉了优选语义，并且其规模关于 $\mathcal{U}$ 也是多项式的。

语义在 DL-PA 中的表示，而此前尚未对这些语义进行刻画（朴素语义、半稳定语义、阶段语义、理想语义和积极语义）；(ii) 我们还将此前的编码调整为适用于更一般的情形（尤其是，我们将完全语义、优选语义和基础语义的编码 [36] 调整为适用于存在论据背景全集 $\mathcal{U}$的假设，而这一假设有助于对论证框架的动态和不确定性进行建模）。其次，我们讨论了此后出现的若干密切相关的研究（第 4.3.2 节和第 4.4 节）。第三，我们给出了关于抽象论证中动态与不确定性相结合的新结果：第 5.2 节和第 5.3 节均为全新内容。此外，与会议论文版本相比，本文还进行了若干细微改进，其中一些已在文中标明。

论证的认知层面。近年来，出现了一些探讨认知逻辑与形式论证相结合的论文。大体而言，这些研究可分为两个主要方向：（i）尝试基于论证工具（如 [44, 62, 63, 61, 21, 22]）对正当信念这一概念进行形式化；（ii）利用认知模型，针对不确定论证框架（如 [60, 59, 49, 50]）进行推理。显然，第二个方向无论在概念上还是技术上，都与本文提出的一些观点密切相关。两者最主要的区别在于，本文使用的形式体系缺乏一种刻画高阶认知态度的工具，也就是说，无法表示主体对其对手的论证情境（即对手的论证框架）的看法，更无法表示主体对对手如何看待该主体的论证情境的看法，依此类推。这一点很重要，因为在策略性论证这一子领域中，这类心理态度已被成功用于递归对手模型（例如，参见 [64]）。然而，要将这种多主体性与一整套动态工具相结合，就意味着要以动态认知逻辑强大的建模能力取代 DL-PA [34]。代价是相关推理任务的计算复杂度会大幅上升。不过，也可以仿效 [27]，采用禁止认知算子辖域内出现析取的轻量级认知逻辑。如此一来，便能在建模多主体性与动态性和建模不确定性之间取得折衷：因为这一方法必须排除所有超出不完全论证框架范围的不确定性（见图2）。第二个重要区别在于，不同于认知逻辑，本文研究的各种形式体系都无法对现实世界进行建模，即刻画独立于形式化代理的想法而为真的内容。反过来说，这一概念是区分知识（通常要求为真）与信念（通常只要求保持一致）所必需的。不过，这一限制似乎较容易克服：只需为不完全论证框架（及其扩展）增添一个特定的补全，以非正式的方式说明现实中的论证框架是什么即可。

进一步研究语义。未来研究的另一个方向，是将我们的 DL-PA 编码扩展到第 3 节尚未考虑的语义。一个特别值得关注的案例是近期提出的基于弱可接受性的语义系列 [10], ，因为已有研究表明，其大多数相关推理任务都是 PSPACE-完全的，与 DL-PA模型检验问题 [5] 的复杂度相同。

另一种表达力概念。最近， [2], Alfano等人提出了一种重写技术，用以将一般的 IAF 化简为其严格子类 arg-IAF 和 att-IAF，甚至化简为 arg-IAF 的一个真子类。[13] 更具体地说，他们证明了 [2, 定理 7] ：重写后的不完全论证框架的补全可以（通过另一种变换）映射到原始框架的补全。他们还声称：“这一结果意味着，arg-IAF（以及 farg-IAF、att-IAF）的表达力与一般 IAF 相同，尽管 arg-IAF（以及 farg-IAF、att-IAF）的结构更简单。”这显然与我们在命题9和图2中给出的表达能力图相冲突；该图基于 [53] 中首次提出、并在 [52, 48]中沿用的表达力概念。尽管详细比较这两种表达力概念超出了本讨论的范围，我们只想指出，这里采用的概念似乎更适合直观建模（即，了解形式化代理能够在头脑中表示哪些情境）；而 Alfano 等人采用的概念则似乎在技术层面更有意义（事实上，它被用于将关于 IAF 的复杂性结果推广到其真子类）。无论如何，[2]中的工作提出了一个有趣的研究问题：这种重写技术能否扩展到表达力更强的形式体系（按我们的定义），例如 rIAF 或 cIAF?

资金支持
两位作者的研究工作均部分获得欧盟 ICT-48 2020年项目 TAILOR（编号：952215）的资助。本研究的部分工作是在 Antonio Yuste 受聘于马拉加大学期间完成的，他通过 Plan Propio de Investigación, Transferencia y Divulgación Científica 的 A.3.1 类博士后合同受聘。

致谢
感谢 Sylvie Doutre 和 Jean-Guy Mailly 就本文主题进行的前期讨论，尤其感谢他们促成了受约束的不完整论证框架这一构想。

[13]所谓的 $fact\text{-}uncertain\ AFs$ （事实不确定论证框架），即所有不确定论证均未受到攻击的论证不完全论证框架。

参考文献

[1] Carlos Alchourr´on、Peter G¨ardenfors 和 David Makinson。《论理论变更逻辑：部分交汇收缩与修订函数》。《符号逻辑杂志》，50：510–530，1985年。[2] Gianvincenzo Alfano、Sergio Greco、Francesco Parisi 和 Irina Trubitsyna。《不完整论证框架：性质与复杂度》。载于《第三十六届AAAI人工智能会议》，第5451–5460页。AAAI出版社，2022年。[3] Leila Amgoud 和 Srdjan Vesic。《一种基于偏好的论证框架新方法》。《数学与人工智能年刊》，63(2)：149–183，2011年。[4] Katie Atkinson、Pietro Baroni、Massimiliano Giacomin、Anthony Hunter、Henry Prakken、Chris Reed、Guillermo Simari、Matthias Thimm 和 Serena Villata。《迈向人工论证》。《人工智能杂志》，38(3)：25–36，2017年。[5] Philippe Balbiani、Andreas Herzig、Fran¸cois Schwarzentruber 和 Nicolas Troquard。《DL-PA 与 DCL-PC：模型检测与可满足性问题确实属于 PSPACE》。CoRR，abs/1411.7825，2014年。[6] Philippe Balbiani、Andreas Herzig 和 Nicolas Troquard。《命题赋值动态逻辑：一种性质良好的 PDL 变体》。载于第28届ACM/IEEE计算机科学逻辑研讨会，第143–152页。IEEE，2013年。[7] Pietro Baroni、Martin Caminada 和 Massimiliano Giacomin。《抽象论证框架及其语义》。载于 Pietro Baroni、Dov M Gabbay、Massimiliano Giacomin 和 Leendert van der Torre 编，《形式论证手册》，第159–236页。College Publications，2018年。[8]Pietro Baroni、Federico Cerutti、Massimiliano Giacomin 和 Giovanni Guida。《抽象论证框架中对攻击的攻击》。载于 Claudio Sossai 和 Gaetano Chemello 编，《欧洲符号与定量推理及不确定性方法会议》，第83–94页。施普林格，2009年。[9] Ringo Baumann 和 Gerhard Brewka。《扩展论证框架：强制与单调性结果》。载于 Pietro Baroni、Federico Cerutti、Massimiliano Giacomin 和 Guillermo Ricardo Simari 编，《COMMA 2010论文集》，人工智能前沿与应用丛书第216卷，第75–86页。IOS Press，2010年。

[10] Ringo Baumann、Gerhard Brewka 和 Markus Ulbricht。重新审视抽象论证的基础——基于弱可接受性和弱防御的语义。载于第三十四届AAAI人工智能会议（AAAI 2020）论文集，第2742–2749页。AAAI出版社，2020年。[11] Ringo Baumann、Sylvie Doutre、Jean-Guy Mailly 和 Johannes P Wall-ner。形式论证中的强制。《应用逻辑杂志》，2631(6):1623，2021年。[12] Ringo Baumann、Dov M. Gabbay 和 Odinaldo Rodrigues。遗忘论证。载于第三十四届AAAI人工智能会议（AAAI 2020）论文集，第2750–2757页。AAAI出版社，2020年。[13] Dorothea Baumeister、Matti J¨arvisalo、Daniel Neugebauer、Andreas Niska-nen 和 J¨org Rothe。不完整论证框架中的接受。人工智能，295:103470，2021年。[14] Dorothea Baumeister、Daniel Neugebauer 和 J¨org Rothe。不完整论证框架中的宽容式接受与怀疑式接受。载于COMMA 2018论文集，人工智能与应用前沿第305卷，第181–192页。IOS Press，2018年。[15]Dorothea Baumeister、Daniel Neugebauer、J¨org Rothe 和 Hilmar Schadrack。不完整论证框架中验证的复杂性。载于Sheila A. McIlraith和Kilian Q. Weinberger编，《第三十二届AAAI人工智能会议论文集》（AAAI-18），第1753–1760页。AAAI出版社，2018年。[16]Dorothea Baumeister、Daniel Neugebauer、J¨org Rothe 和 Hilmar Schadrack。不完整论证框架中的验证。人工智能，264:1–26，2018年。[17] Trevor JM Bench-Capon 和 Paul E Dunne。人工智能中的论证。人工智能，171(10-15):619–641，2007年。[18]Philippe Besnard、Claudette Cayrol 和 Marie-Christine Lagasquie-Schiex。逻辑理论与抽象论证：现有研究综述。《论证与计算》，11(1-2):41–102，2020年。[19] Philippe Besnard 和 Sylvie Doutre。检查一组论证的可接受性。载于J.P. Delgrande和T. Schaub编，《第十届非单调推理国际研讨会》，第59–64页，2004年。[20] Stefano Bistarelli 和 Francesco Santini。ConArg：用于论证系统的基于约束的计算框架。载于IEEE第23届人工智能工具国际会议（I CTAI 2011），第605–612页。IEEE计算机学会，2011年。

[21] Alfredo Burrieza 和 Antonio Yuste-Ginel。认知意识逻辑中关于结构化论证的基本信念与基于论证的信念。载于Henry Prakken、Stefano Bistarelli、Francesco Santini和Carlo Taticchi编，COMMA 2020论文集，人工智能与应用前沿第326卷，第123–134页。IOS Press，2020年。[22] Alfredo Burrieza 和 Antonio Yuste-Ginel。一种关于信念、论证及其动态的认知意识框架。载于Joseph Y. Halpern和Andr´es Perea编，第十八届理性与知识理论问题会议论文集，EPTCS第335卷，第69–83页，2021年。[23] Martin WA Caminada、Walter A Carnielli 和 Paul E Dunne。半稳定语义。《逻辑与计算杂志》，22(5):1207–1254，2012年。[24] Claudette Cayrol、Caroline Devred 和 Marie-Christine Lagasquie-Schiex。论证中的无知处理：部分论证框架的语义。载于Khaled Mellouli编，2007年欧洲符号与定量不确定性推理方法会议论文集，计算机科学讲义第4724卷，第259–270页。施普林格，2007年。[25] Claudette Cayrol 和 Marie-Christine Lagasquie-Schiex。双极论证框架中论证的可接受性。载于Llu´ís Godo编，欧洲符号与定量不确定性推理方法会议论文集，计算机科学讲义第3571卷，第378–389页。施普林格，2005年。[26] Federico Cerutti、Sarah A Gaggl、Matthias Thimm 和 Johannes Wallner。形式论证实现的基础。《IfCoLog逻辑及其应用期刊》，4(8):2623–2705，2017年。[27] Martin C. Cooper、Andreas Herzig、Faustine Maffre、Fr´ed´eric Maris、Elise Perrotin 和 Pierre R´egnier。一种轻量级认知逻辑及其在规划中的应用。人工智能，298:103437，2021年。[28]Sylvie Coste-Marquis、Caroline Devred、S´ebastien Konieczny、Marie-Christine Lagasquie-Schiex 和 Pierre Marquis。Dung论证系统的合并。人工智能，171(10-15):730–753，2007年。[29] Sylvie Coste-Marquis、Caroline Devred 和 Pierre Marquis。受约束论证框架。载于P. Doherty、J. Mylopoulos和C. Welty编，第十届知识表示与推理原理国际会议论文集，第112–122页。AAAI出版社，2006年。[30] Marcos Cramer 和 Leon van der Torre。SCF2——一种针对论证可接受性的理性人类判断的论证语义。载于C. Beierle、M. Ragni、F. Stolzenburg和M. Thimm编，第八届知识与信念动态研讨会论文集，第24–35页。CEUR-WS.org，2019年。

[31] Yannis Dimopoulos、Jean-Guy Mailly 和 Pavlos Moraitis。论证框架的控制。载于 Sheila A. McIlraith 和 Kilian Q. Weinberger（编），《第三十二届AAAI人工智能会议论文集》，第4678–4685页。AAAI出版社，2018年。[32]Yannis Dimopoulos、Jean-Guy Mailly 和 Pavlos Moraitis。基于论证的谈判与不完整的对手画像。载于 N. Agmon、M. E. Taylor、E. Elkind 和 M. Veloso（编），第十八届自主智能体与多智能体系统国际会议（AAMAS 2019），第1252–1260页，2019年。[33] Hans P. van Ditmarsch、Andreas Herzig、J´erˆome Lang 和 Pierre Marquis。内省遗忘。Synthese，169(2)：405–423，2009年。[34] Hans P. van Ditmarsch、Wiebe van der Hoek 和 Barteld Kooi。《动态认知逻辑》。施普林格，2007年。[35] Sylvie Doutre、Andreas Herzig 和 Laurent Perrussel。用于抽象论证的动态逻辑框架。载于 C. Baral、G. De Giacomo 和 T. Eiter（编），《第十四届知识表示与推理原理国际会议》。AAAI出版社，2014年。[36] Sylvie Doutre、Andreas Herzig 和 Laurent Perrussel。动态逻辑中的抽象论证：表示、推理与变化。载于 Beishui Liao、Thomas ̊Agotnes 和 Yi N. Wang（编），《动态、不确定性与推理》，第153–185页。施普林格，2019年。[37] Sylvie Doutre、Faustine Maffre 和 Peter McBurney。用于抽象论证的动态逻辑框架：添加和移除论证。载于 Salem Benferhat、Karim Tabia 和 Moonis Ali（编），《工业、工程及其他领域应用智能系统国际会议》，计算机科学讲义第10351卷，第295–305页。施普林格，2017年。[38] Sylvie Doutre 和 Jean-Guy Mailly。约束与变化：抽象论证动态综述。《论证与计算》，9：223–248，2018年。[39] Phan Minh Dung。论论证的可接受性及其在非单调推理、逻辑程序设计和多人博弈中的根本作用。《人工智能》，77(2)：321–357，1995年。[40] Bettina Fazzinga、Sergio Flesca 和 Filippo Furfaro。重新审视不完整抽象论证框架上的扩展概念。载于 Christian Bessiere（编），《国际人工智能联合会议论文集》，第1712–1718页。IJCAI组织，2020年7月。[41]Bettina Fazzinga、Sergio Flesca 和 Filippo Furfaro。在相关性存在的情况下，对论证不完整的抽象论证框架进行推理。载于周志华（编），《国际人工智能联合会议论文集》，第189–195页。IJCAI组织，2021年。

[42]Bettina Fazzinga、Sergio Flesca 和 Filippo Furfaro。在相关性存在的情况下，对攻击不完整抽象论证框架进行推理。载于 M. Bienvenu、G. Lakemeyer 和 E. Erdem（编），《第18届知识表示与推理原理国际会议论文集》，第301–311页，2021年11月。[43] Davide Grossi。论论证理论的逻辑。载于《第9届自主智能体与多智能体系统国际会议论文集》，第409–416页。IFAMA，2010年。[44] Davide Grossi 和 Wiebe van der Hoek。通过正当论证获得正当信念。载于 Chitta Baral、Giuseppe De Giacomo 和 Thomas Eiter（编），《国际知识表示与推理原理会议》。AAAI出版社，2014年。[45] Mathieu Guillaume、Marcos Cramer、Leendert van der Torre 和 Christine Schiltz。对冲突信息的推理：形式论证的实证研究。PLOS ONE，17(8)：e0273225，2022年。[46] David Harel、Dexter Kozen 和 Jerzy Tiuryn。《动态逻辑》。麻省理工学院出版社，2000年。[47] Andreas Herzig。信念变化操作：用命题赋值动态逻辑讲述一段几乎无所不包的简史。载于 Chitta Baral 和 Giuseppe De Giacomo（编），《国际知识表示与推理原理会议》。AAAI出版社，2014年。[48] Andreas Herzig 和 Antonio Yuste-Ginel。具有定性不确定性的抽象论证：动态逻辑分析。载于 Pietro Baroni、Christoph Benzm¨uller 和 Yi N. W´ang（编），《逻辑与论证》，计算机科学讲义第13040卷，第190–208页。施普林格，2021年。[49] Andreas Herzig 和 Antonio Yuste-Ginel。具有不完整攻击关系知识的多智能体抽象论证框架。载于周志华（编），《国际人工智能联合会议论文集》，第1922–1928页。IJCAI组织，2021年。[50] Andreas Herzig 和 Antonio Yuste-Ginel。论不完整论证框架的认知逻辑。载于 M. Bienvenu、G. Lakemeyer 和 E. Erdem（编），《第18届知识表示与推理原理国际会议论文集》，第681–685页，2021年11月。[51]Anthony Hunter、Sylwia Polberg、Nico Potyka、Tjitze Rienstra 和 Matthias Thimm。概率论证：综述。载于 Dov Gabbay、Massimiliano Giacomin、Guillermo R. Simari 和 Matthias Thimm（编），《形式论证手册》第2卷，第159–236页。College Publications，2021年。

[52] Jean-Guy Mailly。《是、否、也许、我不知道：不完备知识下抽象论证的复杂性与应用》。《论证与计算》，13(3):291–324。[53] Jean-Guy Mailly。《关于丰富型不完备论证框架的注记》。arXiv 预印本 arXiv:2009.04869，2020年。[54] Jean-Guy Mailly。《受约束的不完整论证框架》。载于 Jiˇrina Vejnarov´a 和 Nic Wilson（编），《不确定性推理的符号与定量方法》，计算机科学讲义第12897卷，第103–116页。施普林格，2021年。[55]Sanjay Modgil、Francesca Toni、Floris Bex、Ivan Bratko、Carlos I. Chesˇnevar、Wolfgang Dvoˇr´ak、Marcelo A. Falappa、Xiuyi Fan、Sarah Alice Gaggl、Alejandro J. Garc´ıa、Mar´ıa P. Gonz´alez、Thomas F. Gordon、Joˇao Leite、Martin Moˇzina、Chris Reed、Guillermo R. Simari、Stefan Szeider、Paolo Torroni 和 Stefan Woltran。《论证的附加价值》。载于《协议技术》，第357–403页。施普林格，2013年。[56] Andreas Niskanen。《抽象论证中的动态与不确定性：计算方法》。博士论文，赫尔辛基大学，2020年。[57]Andreas Niskanen 和 Matti J¨arvisalo。 μ-toksia：一种高效的抽象论证推理器。载于 Diego Calvanese、Esra Erdem 和 Michael Thielscher（编），《第17届国际知识表示与推理原理会议论文集》，第800–804页，2020年。[58] Andreas Niskanen、Daniel Neugebauer、Matti J¨arvisalo 等。《控制论证框架的可控性》。载于 Christian Bessiere（编），《国际人工智能联合会议》。IJCAI 组织，2020年。[59] Carlo Proietti 和 Antonio Yuste-Ginel。《用于抽象论证的动态认知逻辑》。《Synthese》，199(3):8641–8700，2021年。[60] Fran¸cois Schwarzentruber、Srdjan Vesic 和 Tjitze Rienstra。《构建用于论证的认知逻辑》。载于 Luis Fariˇnas del Cerro、Andreas Herzig 和 J´erˆome Mengin（编），《人工智能中的逻辑》，计算机科学讲义第7519卷，第359–371页。施普林格，2012年。[61] Chenwei Shi、Sonja Smets 和 Fernando R Vel´azquez-Quesada。《基于论证的正当信念逻辑》。《Erkenntnis》，第1–37页，2021年。[62]Chenwei Shi、Sonja Smets 和 Fernando R. Vel´azquez-Quesada。《拓扑结构中的基于论证的信念》。载于 J´erˆome Lang（编），《第十六届理性与知识的理论方面会议论文集》，EPTCS。开放出版协会，2017年。

[63] Chenwei Shi、Sonja Smets 和 Fernando R. Vel´azquez-Quesada。《由二元论证支持的信念》。《应用非经典逻辑杂志》，28(2-3):165–188，2018年。

[64] Matthias Thimm。《多智能体系统中的策略性论证》。KI- K ¨ünstliche Intelligenz，28(3):159–168，2014年。

附录

在本附录中，我们给出全文各处结果的部分证明和证明概要。

[定理 **1**]

证明。我们只证明 $\sigma = se$ 的第一条。其他情况较为简单，且可用类似的论证推出；第二条则可由第一条和 makeExt 的含义轻易得出。首先，我们不加证明地陈述几个必要的引理：

引理 1。(v, v_1) $\in$ ||copy($\mathrm{IN}_{\mathcal{U}}$); makeExt$^{\sigma}$|| 蕴含 $\mathrm{E}_v = \{x \mid \mathrm{in}'_x \in \mathrm{v}_1\}$ and $(A_v, R_v) = (A_{v_1}, R_{v_1})$。

引理 2. 设 v $\subseteq$ Prp，满足 v $\models$ 良好且 $\{x \mid \mathrm{in}'_x \in \mathrm{v}\} \subseteq \{x \mid \mathrm{aw}_x \in \mathrm{v}\}$，则：

- v $\models$ IncludesCp 当且仅当 $\{x \mid \mathrm{in}'_x \in \mathrm{v}\}^{\oplus} \subseteq \mathrm{E}_v^{\oplus}$。
- v $\models$ IncludedInCp 当且仅当 $\mathrm{E}_v^{\oplus} \subseteq \{x \mid \mathrm{in}'_x \in \mathrm{v}\}^{\oplus}$。

(⇒) 假设 v $\models$半稳定，这等价于 v $\models$完全且 v $\models$ [copy($\mathrm{IN}_{\mathcal{U}}$); makeExt^{co}] (IncludesCp $\rightarrow$ IncludedInCp)。根据我们正在证明的同一结论（但针对 $\sigma = co$），第一个合取项等价于 $\mathrm{E}_v \in$ co(A_v, R_v)。因此，我们只需证明 E_v 在完备扩展中具有极大范围。假设 $E' \in$ co(A_v, R_v)。根据我们正在证明的同一结论（但针对 $\sigma = co$ ）和引理 1，可得 $E' = \mathrm{E}_{v_1}$ 且对于某个 (v, v_1)，$\mathrm{E}_v = \{x \mid \mathrm{in}'_x \in \mathrm{v}_1\} \in$ ||copy($\mathrm{IN}_{\mathcal{U}}$); makeExt^{co}||。假设 $\mathrm{E}_v^{\oplus} \subseteq \mathrm{E}_{v_1}^{\oplus}$。注意，$\mathrm{v}_1$ 满足引理 2 的前提（这可由 v $\models$ 完全和引理 1 推出）。因此，$\mathrm{E}_v^{\oplus} \subseteq \mathrm{E}_{v_1}^{\oplus}$ 等价于 $\mathrm{v}_1 \models$ IncludesCp。由于我们知道 $\mathrm{v}_1 \models$ IncludesCp $\rightarrow$ IncludedInCp，因此可推出 $\mathrm{v}_1 \models$IncludedInCp；而根据引理 2，这又等价于 $\mathrm{E}_{v_1}^{\oplus} \subseteq \mathrm{E}_v^{\oplus}$。由于 E_{v_1} 是 (A_v, R_v) 的任意完备扩展，我们可以得出结论：E_v 的范围在所有完备扩展的范围中是极大的。

(⇐) 假设 $\mathrm{E}_v \in$ se((A_v, R_v))，这相当于：(i) $\mathrm{E}_v \in$ co(A_v, R_v)；(ii) E_v的范围在 (A_v, R_v) 的完全扩展中是极大的。由 (i) 以及我们正在证明的同一条结论（但针对 $\sigma = co$），可得 v $\models$ 完全。因此，我们只需证明半稳定语义的第二个合取项在 v 处成立。为此，假设 (v , v_1) $\in$ ||copy($\mathrm{IN}_{\mathcal{U}}$); makeExt^{co}|| 且 $\mathrm{v}_1 \models$IncludesCp。由这两个引理和前述断言，可得 $\mathrm{v}_1 \models$IncludedInCp。

□

[命题 **2**]

证明。对于第一项，假设 ($\mathrm{v}_{\mathsf{IAF}}$, v) $\in$ ||makeComp^{IAF}||。根据命题 1，我们回顾到，对于任意原子集合 P，||mkTrueSome(P)|| = {(v′, v″) | v″ = v′ $\cup S, S \subseteq$ P}。根据顺序组合算子";"的语义，($\mathrm{v}_{\mathsf{IAF}}$, v) $\in$ ||makeComp^{IAF}|| 意味着，对于某个 P $\subseteq \mathrm{AW}_{A^?} \cup \mathrm{ATT}_{R^?}$，v = $\mathrm{v}_{\mathsf{IAF}} \cup$ P。由此陈述，并应用 (A_v, R_v) 的定义和补全的定义，我们得到 (A_v, R_v) $\in$ completions(IAF)。

对于第二项，假设 (A^*, R^*) $\in$ 完全扩展(IAF)，这相当于 $A^F \subseteq A^* \subseteq A^F \cup A^?$ 和 $R^F \restriction_{A^*} \subseteq R^* \subseteq (R^F \cup R^?) \restriction_{A^*}$。现在，记住 $\mathrm{v}_{(A^*,R^*)} = \mathrm{AW}_{A^*} \cup \mathrm{ATT}_{R^*}$。根据前面两个陈述以及 $\mathrm{v}_{\mathsf{IAF}}$ 的定义，我们可以推断出，其真值与 v 不同于 $\mathrm{v}_{(A^*,R^*)}$ 的变量集合必须是 $\mathrm{AW}_{A^?} \cup \mathrm{ATT}_{R^?}$ 的子集；正如前文所述，这相当于说 ($\mathrm{v}_{\mathsf{IAF}}$, $\mathrm{v}_{(A^*,R^*)}$) $\in$ ||makeComp^{IAF}||。 □

[命题 **3**]

证明概要。该结果可由推理任务的定义、每个 makeExt$^{\sigma}$ (定理 1) 的正确性、命题2，以及 DL-PA 中 [.] 和 ⟨.⟩ 的解释推出。

□

[命题 **4**]

证明概要。证明与命题 2 的证明类似，但考虑了以下观察：当应用于对称关系 $R^{\leftrightarrow}$ ={(x_1, y_1)、(y_1, x_1) ...、(x_n, y_n)、(y_n, x_n)} 时，对于每个 $1 \leq i \leq n$，dis($\mathrm{ATT}_{R^{\leftrightarrow}}$) 的每次执行都会使 r_{x_i,y_i} 或 r_{y_i,x_i} 为真，或者二者均为真。这确保了 rIAF 补全定义的最后一个子句在 DL-PA 程序 makeComp^{rIAF} 中得以体现。

□

[命题 **5**]

证明概要。该结果直接由推理问题的定义、makeExt$^{\sigma}$ 的正确性（定理 1）、makeComp^{rIAF} 的正确性（命题 4）以及 DL-PA 的语义得出。

□

[命题 **6**]

证明概要。将 vary(AW_A); vary($\mathrm{ATT}_{A\times A}$) 的解释限制在 $2^{\mathrm{Prp}_A \setminus \mathsf{IN}^A}$ 上，实际上就是全关系 $2^{\mathrm{Prp}_A \setminus \mathsf{IN}_A} \times 2^{\mathrm{Prp}_A \setminus \mathsf{IN}^A}$。因此，从 $\mathrm{v}_{\mathsf{cIAF}} = \emptyset$ 出发，vary(AW_A); vary($\mathrm{ATT}_{A\times A}$) 可以执行到 $2^{\mathrm{Prp}_A \setminus \mathsf{IN}^A}$ 中的任意赋值。随后执行 φ?，筛选出 $2^{\mathrm{Prp}_A \setminus \mathsf{IN}^A}$ 中满足 cIAF 约束的那些赋值，即表示 cIAF 补全集合的 $2^{\mathrm{Prp}_A \setminus \mathsf{IN}^A}$ 中的赋值集合。

□

[命题 **7**]

证明概要。该结果可由推理任务的定义、makeExt$^{\sigma}$ 的正确性（定理 1）、命题 6 以及 DL-PA 的语义得出。

□

[命题 **8**]

证明。我们只需证明 $c\text{-}\mathcal{IAF} \succ \mathcal{RIAF}$，因为 $c\text{-}\mathcal{IAF} \succ \mathcal{IAF}$ 可由 $\mathcal{RIAF} \succ \mathcal{IAF}$ [53] 以及 $\succ$的传递性推出。

为证明 $c\text{-}\mathcal{IAF} \succeq \mathcal{RIAF}$，设 rIAF 是一个具有补全(rIAF) $=\{(A_1^*, R_1^*)$ ...、$(A_n^*, R_n^*)\}$ 的 rIAF。对于定义在 $\mathcal{U}$ 上的任意论证框架 (A, R)，我们可以写出其理论（例如参见 [37]），即命题公式

$$\mathsf{Th}(A,R) = \bigwedge_{x\in A} \mathtt{aw}_x \wedge \bigwedge_{x\in\mathcal{U}\setminus A} \neg\mathtt{aw}_x \wedge \bigwedge_{(x,y)\in R} \mathtt{r}_{x,y} \wedge \bigwedge_{(x,y)\in\mathcal{U}\times\mathcal{U}\setminus R} \neg\mathtt{r}_{x,y}.$$

接着很容易证明，对于任意赋值 v $\subseteq$ Prp，都有 v $\models$Th(A, R) 当且仅当 $(A_v, R_v) = (A, R)$。现在令 $\rho = \bigvee_{1\le i\le n} \mathsf{Th}(A_i^*, R_i^*)$，我们有

$$\mathsf{completions}(\mathcal{U}, \rho) = \mathsf{completions}(\mathsf{rIAF}).$$

为了证明 $\mathcal{RIAF} \not\succeq c\text{-}\mathcal{IAF}$，只需说明例 3 的 cIAF（称为 cIAF$_0$）无法表示为 rIAF。为导出矛盾，假设存在一个 rIAF rIAF $= (A^F, R^F, A^?, R^?, R^{\leftrightarrow})$，其补全集合与 cIAF$_0$相同。那么我们将有 $(a, b) \in R^F \cup R^? \cup R^{\leftrightarrow}$（因为 (a, b) 出现在 rIAF 的某个补全中）。我们将证明最后一个陈述是荒谬的。如果 $(a, b) \in R^F$，那么在 a 和 b 都存在的情况下，(a, b) 应出现在 rIAF 的所有补全中，但事实并非如此。如果 $(a, b) \in R^?$，那么我们根据 $(b, a) \in R^F \cup R^? \cup R^{\leftrightarrow}$ 分情况讨论：第一种情况是不可能的，因为在 a 和 b 出现的每个补全中，(b, a) 都应当存在，而事实并非如此；第二种情况是荒谬的，因为我们将得到一个既不包含 (a, b) 也不包含 (b, a) 的扩展，而事实并非如此；第三种情况是不可能的，因为它将意味着 $(a, b) \in R^{\leftrightarrow}$，但我们已假设 $(a, b) \in R^?$，并且根据定义我们知道 $R^? \cap R^{\leftrightarrow} = \emptyset$。最后，假设 $(a, b) \in R^{\leftrightarrow}$，这意味着 $(b, a) \in R^{\leftrightarrow}$（由 $R^{\leftrightarrow}$ 的对称性），这是不可能的，因为我们将得到一个同时包含 (a, b) 和 (b, a) 的补全，而事实并非如此。

□

[命题 **10**]

证明概要。证明与命题2和命题4的证明类似。关键区别在于，需要先执行 control^{CAF}，以非确定性地选择 CAF 的一个控制配置。还需注意，$\mathsf{ATT}_{R^C} \subseteq \mathsf{v}_{\mathsf{CAF}}$ 对于在相应补全中获得所需的控制攻击至关重要。

□

[命题 **11**]

证明概要。该结果由推理任务的定义、makeExt$^{\sigma}$ 的正确性（定理 1）、命题 10以及 DL-PA 的语义直接得出。

□