

# 論理型言語 PROLEG から双極議論フレームワークへの変換の Coq による形式化と証明

中西 瑠海<sup>†1</sup> 森口 草介<sup>†2</sup> 高橋 和子<sup>†3</sup>

<sup>†</sup> 関西学院大学 理工学部

## 1 はじめに

計算による法的推論を行うシステムとして、論理型言語 PROLEG [4] が考案されている。PROLEG では、条文を論理プログラムで記述することで推論過程が明確になる一方で、そのような記述が法曹に親しみにくいという問題点があった。そこで川崎らは意味を保持したまま、グラフ表現が可能な双極議論フレームワーク (BAF) に変換する手法を提案した [3]。この変換の正当性、すなわち変換後の BAF が元のプログラムの意味を保持することは既に手証明が与えられている。しかし、プログラム変換においては小さな誤りが深刻なエラーにつながることもあるため、機械的な証明により厳密性を保証することが望ましい。本研究の目的は、[3] における変換の正当性を証明支援系 Coq [2] を用いて証明することである。

## 2 論理型言語 PROLEG

PROLEG は Prolog を拡張して開発された、計算による法的推論を行うためのシステムである [4]。

**定義 1.** PROLEG プログラム  $P$  は規則の有限集合  $\mathcal{R}$  と例外の有限集合  $\mathcal{E}$  を用いて  $P = \langle \mathcal{R}, \mathcal{E} \rangle$  の 2 つ組で定義される。

PROLEG プログラムの例を示す。  $p, q_1, q_2, r$  はアトムである。

**例 1.**

$$p \leq q_1, q_2.$$

$$\text{exception}(q_1, r).$$

$$q_2 \leq .$$

$$r \leq .$$

$p \leq q_1, q_2.$  は構成要件を述べたもの (規則) であり、 $q_1, q_2$  が成り立つならば  $p$  が導出されることを意味する。  $p$  をこの規則の *head*,  $\{q_1, q_2\}$  を *body* と呼ぶ。一方、 $\text{exception}(q_1, r).$  は例外的なルール (例外) であり、 $r$  が成り立つならば  $q_1$  が必ず偽となることを意味する。

### Formalization of the Transformation from PROLEG to a Bipolar Argumentation Framework Using Coq

Ruka NAKANISHI<sup>†</sup>, Sosuke MORIGUCHI<sup>†</sup> and Kazuko TAKAHASHI<sup>†</sup>

<sup>†</sup>School of Science and Technology, Kwansei Gakuin University  
2-1 Gakuen, Sanda, Hyogo, 669-1337, Japan  
{fms49117<sup>1</sup>, ktaka<sup>3</sup>}@kwansei.ac.jp, chiguri@acm.org<sup>2</sup>

規則と同様に、 $q_1$  をこの例外の *head*,  $\{r\}$  を *body* と呼ぶ。  $q_2 \leq .$  や  $r \leq .$  は *body* が空の規則であり、事実であることを意味する。

PROLEG プログラムの意味論は解集合によって与えられる [4]。

**定義 2.** アトムの集合  $M$  が  $\{H \leq B_1, \dots, B_n \in \mathcal{R} \mid \forall E \in \mathcal{E}, \text{if } \text{head}(E) = H \text{ then } \text{body}(E) \not\subseteq M\}$  の最小モデルであるとき、 $M$  を  $\langle \mathcal{R}, \mathcal{E} \rangle$  の解集合と呼ぶ。

例 1 のプログラムでは解集合は  $\{q_2, r\}$  となる。

また、 $P$  中の各アトム  $A$  に対して関連集合  $r_P(A)$  およびレベル  $l_P(A)$  が次のように定義されている [3]。  $r_P(A)$  はアトム  $A$  の導出に関係するアトムの集合である。レベル関数  $l_P$  では  $A$  のレベルを  $r_P(A)$  の各アトムよりも大きくすることでアトムに順序を付けている。

**定義 3.**  $r_P(A) = \bigcup_{R \in \mathcal{R} \wedge \text{head}(R) = A} \text{body}(R) \cup \bigcup_{E \in \mathcal{E} \wedge \text{head}(E) = A} \text{body}(E)$

**定義 4.**  $l_P(A) = \begin{cases} 1 & (r_P(A) = \emptyset) \\ \max_{B \in r_P(A)} l_P(B) + 1 & (\text{otherwise}) \end{cases}$

$l_P$  が一意に定まるとき、 $P$  は非循環であるという。本研究では非循環な PROLEG プログラムのみを考える。

## 3 双極議論フレームワーク (BAF)

川崎らは Cayrol らが定義した BAF [1] を拡張し、次のように定義した [3]。

**定義 5.** 双極議論フレームワークは  $\langle AR, ATT, SUP \rangle$  の 3 つ組で定義される。ここで  $AR$  は論証の有限集合、 $ATT \subseteq AR \times AR$  は論証間の攻撃関係、 $SUP \subseteq (2^{AR} \setminus \{\emptyset\}) \times AR$  は論証の集合と論証との間の支持関係である。

BAF の意味論は、論証の受理性を表すラベリングによって与えられる。川崎らは、PROLEG プログラムを変換して得られる BAF がもとのプログラムの意味を保持するようにラベリングを定義した。

**定義 6.** 双極議論フレームワーク  $\langle AR, ATT, SUP \rangle$  について、ラベリング  $\mathcal{L}$  は  $AR$  から  $\{in, out\}$  への関数であり、任意の  $A \in AR$  に対して以下を満たすとき完全ラベリングであるという。

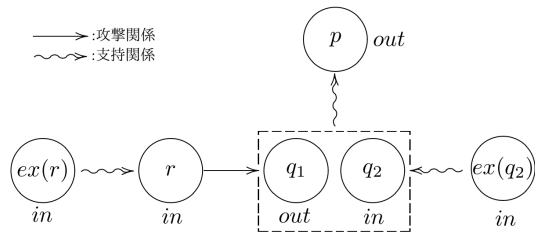


図 1: 変換後の BAF の完全ラベリング付きグラフ表現

- $(\forall B \in AR, (B, A) \notin ATT) \wedge (\forall S \subseteq AR, (S, A) \notin SUP) \Rightarrow \mathcal{L}(A) = in$
- $(\forall B \in AR, (B, A) \in ATT \Rightarrow \mathcal{L}(B) = out) \wedge (\exists S \subseteq AR, (S, A) \in SUP \wedge (\forall X \in S, \mathcal{L}(X) = in)) \Rightarrow \mathcal{L}(A) = in$
- otherwise  $\mathcal{L}(A) = out$

#### 4 PROLEG から BAF への変換

川崎らは法的推論を一種の議論とみなすことで、次のように PROLEG プログラムを BAF に変換した。

- アトムを BAF における論証とみなす。
- 規則を BAF における支持関係に変換する。
- 例外を BAF における攻撃関係に変換する。
- $q_2 \leq .$  のように事実を表す規則については存在論証  $ex(q_2)$  を  $AR$  に追加し、 $ex(q_2)$  によって  $q_2$  を支持するものとする。
- *head* として一度も現れていないアトム  $A$  は不在論証  $ab(A)$  を  $AR$  に追加し、 $ab(A)$  によって攻撃されているものとする。

この変換により、例 1 のプログラムは次のような BAF に変換される。

$$\langle \{p, q_1, q_2, r, ex(q_2), ex(r)\}, \{(r, q_1)\}, \{(\{q_1, q_2\}, p), (\{ex(q_2)\}, q_2), (\{ex(r)\}, r)\} \rangle$$

図 1 にこの BAF の完全ラベリング付きのグラフ表現を示す。ラベルが *in* となる論証の集合は、存在論証と不在論証を除くと、もとのプログラムの解集合と一致していることがわかる。一般に、次の定理が成り立つことの手証明が [3] で与えられている。

**定理 1.** プログラム  $P$  を変換して得られた BAF を  $baf$  とし、 $M$  を  $P$  の解集合、 $\mathcal{L}$  を  $baf$  の完全ラベリングとする。 $P$  中の任意のアトム  $H$  について、 $\mathcal{L}(H) = in$  ならばそのときに限って  $H \in M$  である。

#### 5 Coq による形式化と証明

PROLEG プログラム, BAF, PROLEG プログラムから BAF への変換を Coq でモデル化し、定理 1 を機械的に証明した。

Theorem preservation:  
`acyclic_program atom_eq_dec p ->`  
`forall H:atom, In H M <-> L (Atom H)=i_n.`

証明は [3] の手証明をベースに補題を作成し、アトムのレベル  $n$  に関する帰納法と場合分けによって行った。Coq での証明に際しては、次のような点を補う必要があった。

手証明では帰納法の初期段階が  $n = 1$  となっているが、Coq においては 0 も自然数であるから、 $n = 0$  の場合の証明を求められる。そのため、最初に任意のアトム  $A$  について  $l_P(A) \neq 0$  であることを証明する必要があった。 $n = 1$  のときはアトムが事実であるかあるいは *head* として現れないアトムであるという場合分けにより証明が進行するため、 $l_P(A) = 1 \Rightarrow (A \text{ は事実}) \vee (A \text{ は head として現れない})$  という補題を証明する必要があった。他にも、規則・例外に対応する支持関係・攻撃関係が存在することや存在論証・不在論証のラベルが *in* になることを補題として証明した。また、帰納段階において規則を支持関係に変換する際、規則の *body* が空でない場合には通常的支持関係、空の場合には存在論証による支持関係となるため場合分けをして考える必要があるが、手証明では空の場合が抜けていたことが分かった。

#### 6 まとめ

非循環な PROLEG プログラムについて、BAF への変換が正しく行われていることを Coq を用いて証明した。非循環な PROLEG プログラムを変換して得られる BAF には唯一の完全ラベリングが存在するが、本研究ではこれを仮定した上で正当性の証明を行ったため、この点も含めた証明を行うことが今後の課題である。また、循環を含む PROLEG プログラムの変換に関する正当性の証明も行う予定である。

#### 参考文献

- [1] Cayrol, C. and Lagasquie-Schieux, M-C.: Bipolar abstract argumentation systems, Artificial Intelligence, pp. 65–84, 2005.
- [2] Coq Proof Assistant, <https://coq.inria.fr/> (2019/01/11 閲覧).
- [3] 川崎 樹, 森口 草介, 高橋 和子: 論理型言語 PROLEG に対応する双極議論フレームワークの意味論について, 日本ソフトウェア科学会第 35 回大会, 2018.
- [4] Satoh, K. et al.: On Generality of PROLEG Knowledge Representation. In *Proc. of JURISIN2012*, pp. 115–128, 2012.