双極議論フレームワーク上での推論システム

高橋 和子

関西学院大学理工学部

2020 年 10 月 26 日
基盤 S 2020 年度前期報告会

目的

- 研究目的: 議論フレームワークを用いて議論の構造を可視化することで推論過程や戦略を明確にし，法曹の理解を助けるとともに，一般関係者に対する説明を明確につくる．

これまでの取り組み

- PROLEG から双極議論フレームワーク (BAF) への変換 [SAFA18, JSSST18, IPSJ19]
- BAF 上での法律推論 [JURISIN18, JSAI19, SUM19, FPAI19]
- PROLEG ブロック図の争点整理バージョンの開発 (佐藤教授と共同, FPAI20)
- 新田教授関係 [JSAI20, etc.]

本発表の概要

- 双極議論フレームワーク
- BAF 上の双方向推論システム
- 検討中の課題
- おわりに

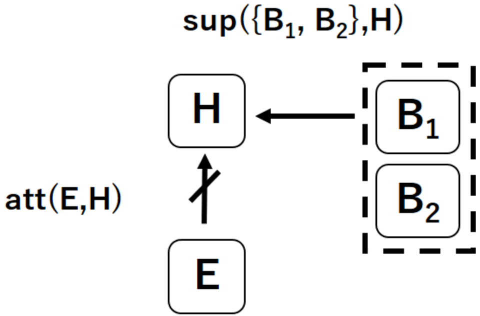
アウトライン

- 双極議論フレームワーク
- BAF 上の双方向推論システム
- 検討中の課題
- おわりに

双極議論フレームワーク (BAF)

双極議論フレームワーク (Bipolar Argumentation Framework,BAF)：議論を攻撃関係と支持関係を用いて構造化して抽象的に表現した枠組み

BAF $= \langle$ 論証集合, 攻撃関係, 支持関係 $\rangle$



$\nleftarrow$ 攻撃関係, $\leftarrow$ 支持関係

法律推論のための双極議論フレームワーク (BAF)

定義

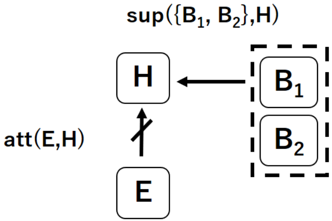
BAF :$\langle AR, ATT, SUP \rangle$
AR: a set of arguments
$ATT \subseteq AR \times AR, \quad SUP \subseteq (2^{AR} \backslash \emptyset) \times AR$

- 支持は論証の集合と論証との関係
- ここではループを含まないもののみを対象

PROLEG プログラムと BAF の対応

PROLEG プログラム		BAF
アトム	$\Leftrightarrow$	論証
例外（exception(H, E).）	$\Leftrightarrow$	攻撃関係（att(E, H)）
規則（$H <= B_1, \ldots, B_n.$）	$\Leftrightarrow$	支持関係（sup$(\{B_1, \ldots, B_n\}, H)$）
事実規則（$H <= .$）	$\Leftrightarrow$	

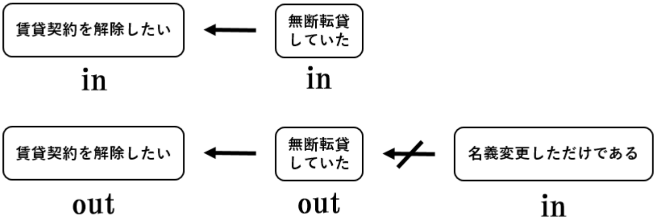
$H <= B_1, B_2.$
exception$(H,\ E)$.



意味論：ラベリング

ラベリング $\mathcal{L} : AR \to \{in, out\}$

- 論証 A が受理可能：$\mathcal{L}(A) = in$
- 論証 A が受理不可能：$\mathcal{L}(A) = out$

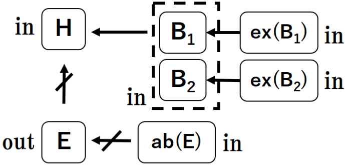


意味論：ラベリング

定義 (完全ラベリング)

$\langle AR, ATT, SUP\rangle$ において，任意の $A \in AR$ に対するラベリング $\mathcal{L}$ が以下を満たす時、完全ラベリングという．

1. $(\forall B \in AR, \neg\mathrm{att}(B, A)) \wedge (\forall \mathrm{A} \subseteq AR, \neg\mathrm{sup}(\mathrm{A}, A))$ であるならば $\mathcal{L}(A) = in$.
2. $(\forall B \in AR, \mathrm{att}(B, A) \Rightarrow \mathcal{L}(B) = out) \wedge (\exists \mathrm{A} \subseteq AR, \mathrm{sup}(\mathrm{A}, A) \wedge \mathcal{L}(\mathrm{A}) = in)$ であるならば，$\mathcal{L}(A) = in$.
3. そうでなければ $\mathcal{L}(A) = out$.



意味論の特徴

法律に対応する BAF

- 法律は一般規則と例外で記述でき，一般規則の条件が成り立つかつ例外が成り立たなければその法律は有効である．
- 条件を論証，一般規則を支持関係，例外を攻撃関係として BAF を構築

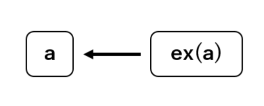
一般の BAF 意味論との相違

- １つでも条件が満たされないと法律は有効ではない ⇒ 支持関係は論証のベキ集合と論証の集合の関係として定義
- 例外のみで定義される法律はなく，必ず一般規則が存在する ⇒ 支持してくれる論証をもたないような論証のラベルは (たとえ out とラベリングされた論証から攻撃されても) out である．

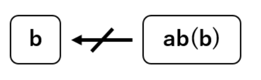
存在論証集合

定義 (存在論証集合)

1. 存在論証 ex(A): 論証 A に対する証拠が存在することを表す論証



2. 不存在論証 ab(A): 論証 A に対する証拠が存在しないことを表す論証



3. 無矛盾な集合: 存在論証の集合 S に対して, $ex(A) \in S$ かつ $ab(A) \in S$ を満たす論証 A がない

アウトライン

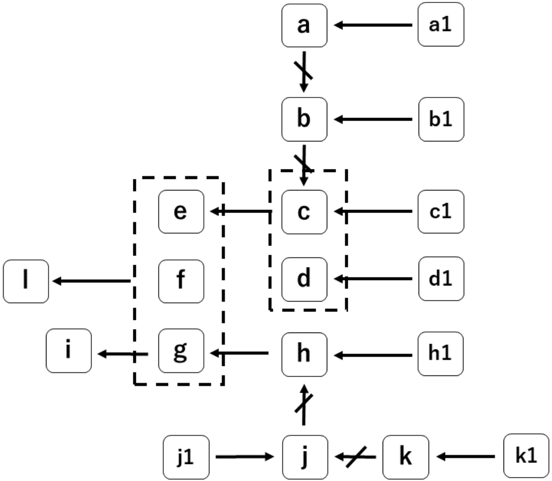
- 双極議論フレームワーク
- BAF 上の双方向推論システム
- 検討中の課題
- おわりに

BAFを用いた法律推論

- ユーザがシステムと対話することで BAF を段階的に構築する．
- 目標とする応用
  - 司法修習生：判決裁判に向けての戦略の確認
  - 一般人：法律相談前の調査検討
- 双方向推論システム
  - 法律の規則部分は BAF として与えられるものとする (ubaf と呼ぶ)（ただしユーザは知らない）
  - 現在わかっている事実 (末端ノード) からはじめてボトムアップに導出可能な論証を導出し，どのような罪状を訴えることができるかを推論
  - それ以外の論証を導出 (罪状を適用) させるためにどのような証拠が必要かをトップダウンに推論

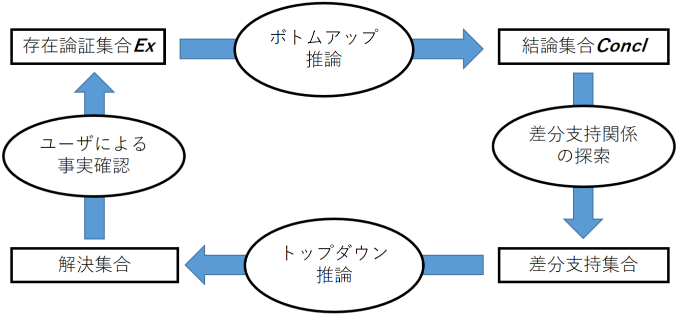
法律全体を表す BAF

法律全体を表す BAF（*ubaf*）の例



注意：支持するノードが 1 つの場合は枠（点線）を省略.

双方向推論：流れ

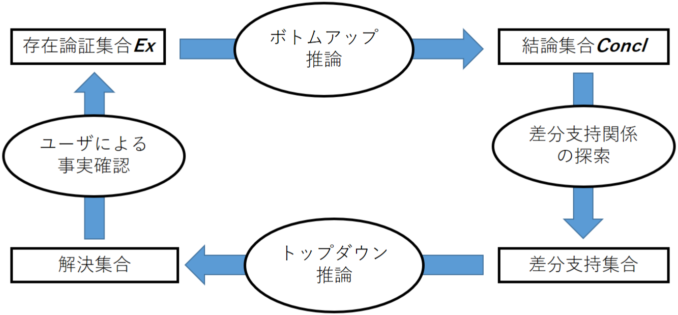


存在論証集合 *Ex*
ボトムアップ推論
結論集合 *Concl*
差分支持関係の探索
差分支持集合
トップダウン推論
解決集合
ユーザによる事実確認

双方向推論：流れ

1. 手持ちの事実集合から支持関係をボトムアップに可能な限りたどり，得られたノードに攻撃関係を追加してラベル付けし，最上流で受理可能な集合を結論集合とする．
2. 新たな上流ノードが得られなくなれば停止する．
3. 結論集合のノードの１つ に注目し，それにいくつかのノードを新たに加えると (差分支持集合)，さらに別の上流のノードを導出できるかどうかを調べる．
4. このようなノードが存在すれば，新たに加えるノードを成り立たせるために必要な解決集合をトップダウンに推論する．
5. 得られた結果をユーザに確認し存在論証集合を更新する．

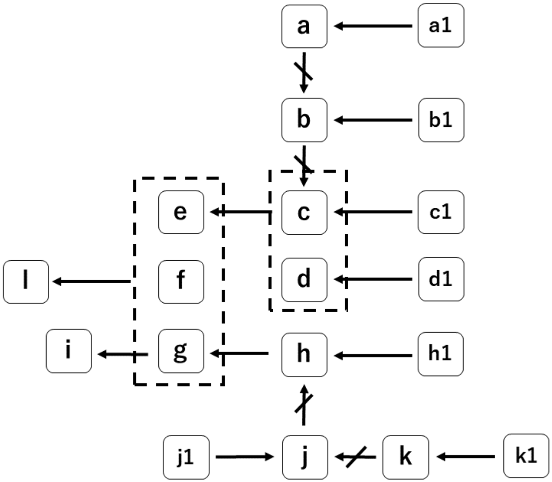
双方向推論：流れ [再掲]



例をあげて説明する.

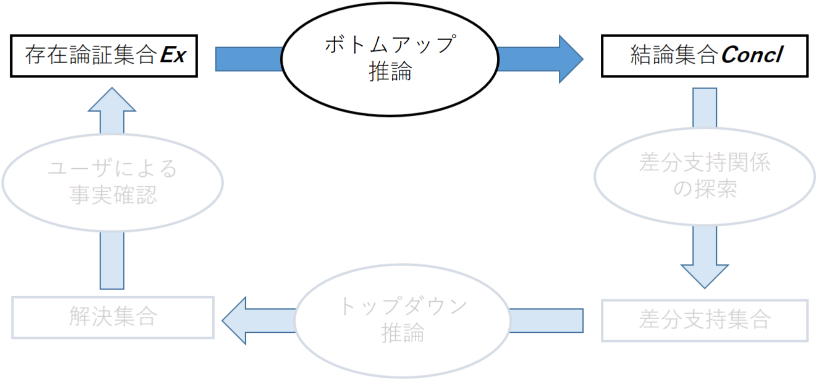
法律全体を表す BAF [再掲]

法律全体を表す BAF（*ubaf*）の例



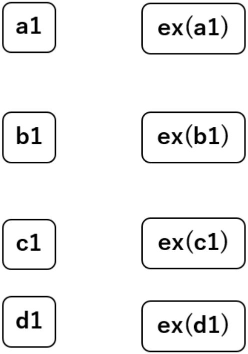
双方向推論

存在論証の集合 $Ex = \{\text{ex(a1), ex(b1), ex(c1), ex(d1)}\}$

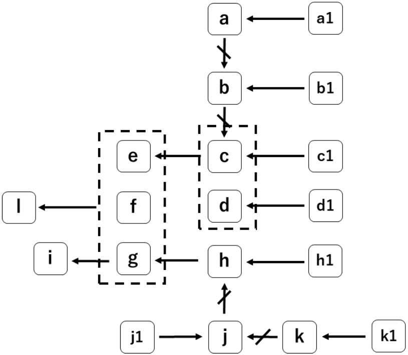


双方向推論：ボトムアップ推論

ボトムアップ推論 ··· 存在論証を与えたとき得られる結論を推論



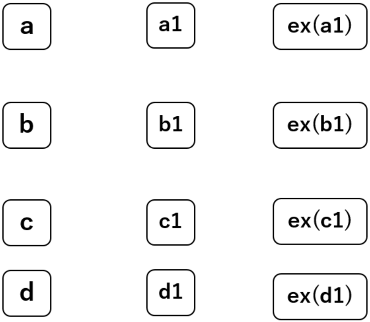
AR(論証の集合)



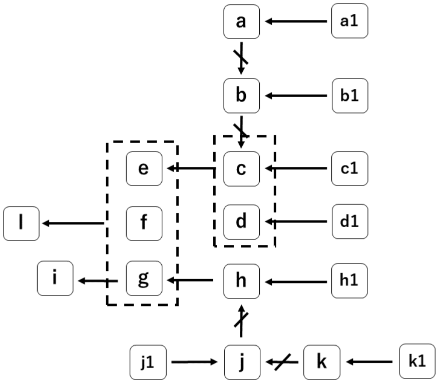
ubaf

双方向推論：ボトムアップ推論

支持関係によって推論が可能な論証を AR に追加



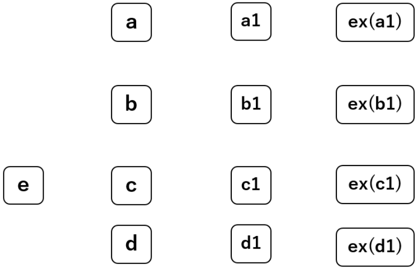
AR



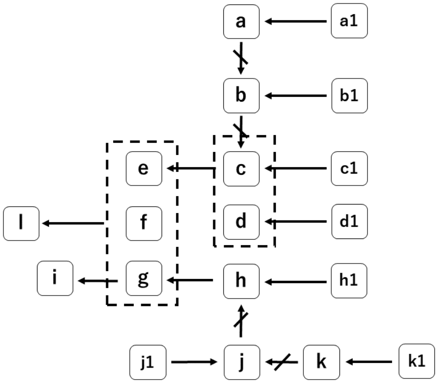
ubaf

双方向推論：ボトムアップ推論

支持関係によって推論が可能な論証を AR に追加



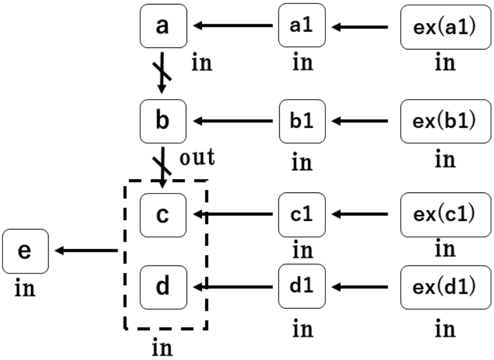
AR



ubaf

双方向推論：ボトムアップ推論

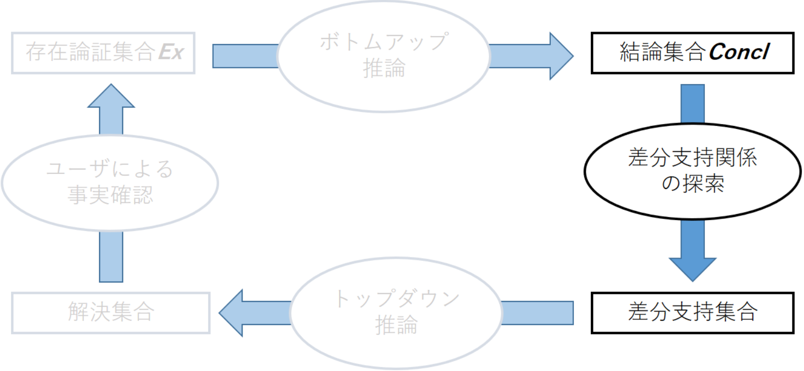
結論集合 *Concl* $\cdots$ 完全ラベリングを与えたとき，ラベルが *in* かつ他の論証を支持していない論証の集合



結論集合 $\mathit{Concl}(\mathit{Ex})=\{\mathrm{a,e}\}$

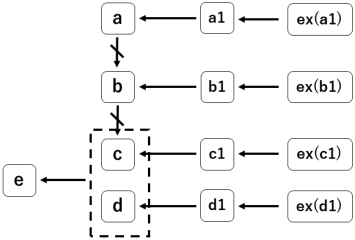
双方向推論

結論集合 $Concl(Ex)$={a,e}

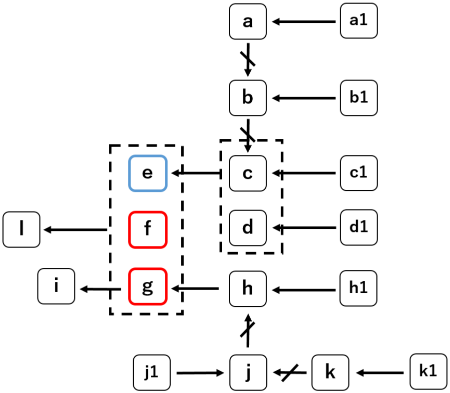


双方向推論：差分支持集合

差分支持集合 $\cdots$ 新たに結論を得る可能性がある論証の集合



BAF

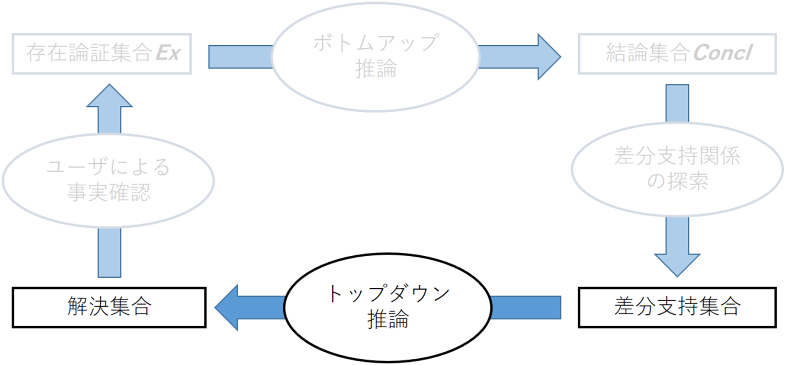


ubaf

差分支持集合 $= \{f, g\}$

双方向推論

差分支持集合 $= \{f, g\}$

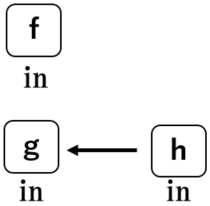


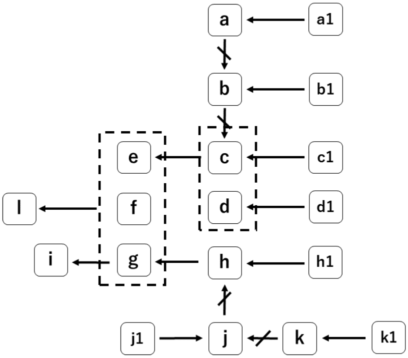
双方向推論：トップダウン推論

トップダウン推論 $\cdots$ 差分支持集合のラベルが *in* であるために必要な（不）存在論証の集合（解決集合）を推論

$\Rightarrow$ 攻撃している論証/支持している論証集合のラベルの条件を推論

差分支持集合 $\{f, g\}$

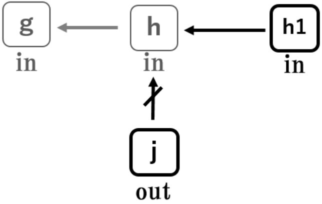


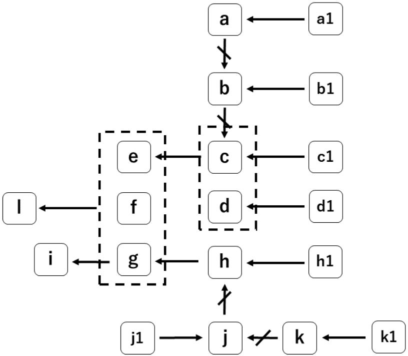


ubaf

双方向推論：トップダウン推論

攻撃している論証/支持している論証集合のラベルの条件を推論

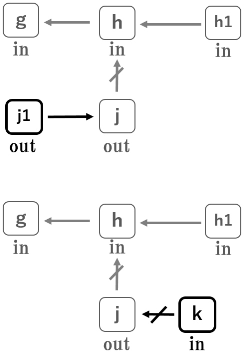


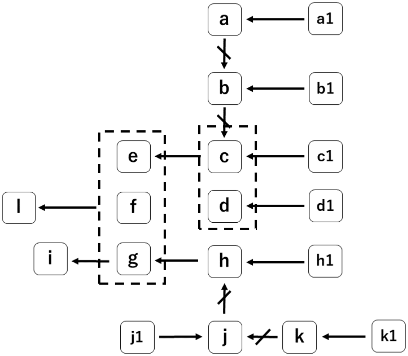


ubaf

双方向推論：トップダウン推論

攻撃している論証/支持している論証集合のラベルの条件を推論



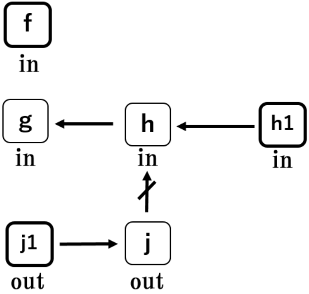


ubaf

双方向推論：トップダウン推論

葉ノードについて

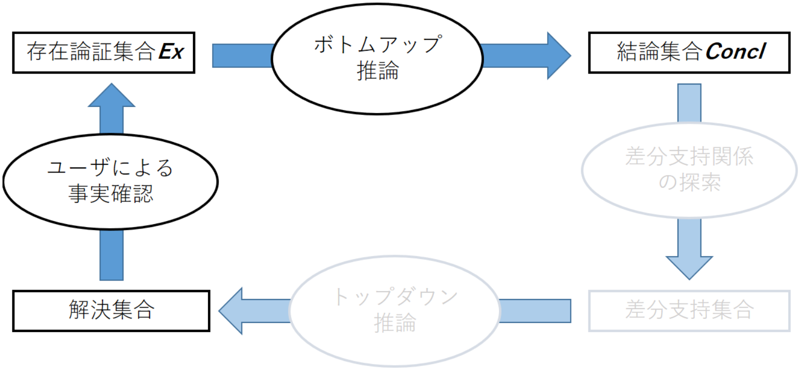
- ラベルが *in* である必要があるならば，解決集合に存在論証を追加
- ラベルが *out* である必要があるならば，解決集合に不存在論証を追加



解決集合,$Sol(\{f\}) = \{\mathrm{ex}(f)\}$,$Sol(\{g\}) = \{\mathrm{ex}(h1), \mathrm{ab}(j1)\}$

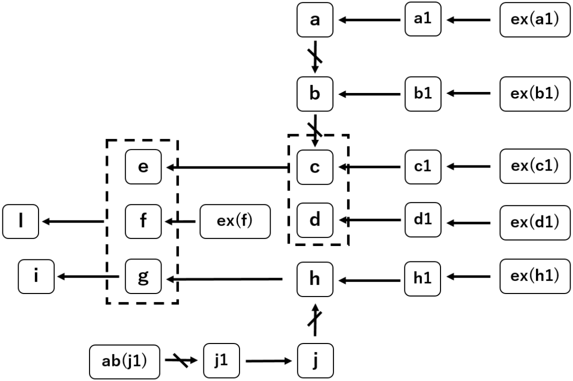
双方向推論

解決集合 $Sol(\{f, g\}) = \{\mathrm{ex}(f), \mathrm{ex}(h1), \mathrm{ab}(j1)\}$
$\Rightarrow$ ユーザにより（不）存在論証の事実確認をして Ex を更新
$Ex = \{\mathrm{ex}(a1), \mathrm{ex}(b1), \mathrm{ex}(c1), \mathrm{ex}(d1), \mathrm{ex}(f), \mathrm{ex}(h1), \mathrm{ab}(j1)\}$



双方向推論：2 周目

$Ex = \{\mathrm{ex}(a1), \mathrm{ex}(b1), \mathrm{ex}(c1), \mathrm{ex}(d1), \mathrm{ex}(f), \mathrm{ex}(h1), \mathrm{ab}(j1)\}$
更新された Ex をもとに再度ボトムアップ推論



demo

アウトライン

- 双極議論フレームワーク
- BAF 上の双方向推論システム
- 検討中の課題
- おわりに

ボトムアップ推論における攻撃関係

ボトムアップ推論では攻撃関係は辿らないので（支持関係に辿ったノード同士の攻撃関係は考慮する）BAF に非連結な部分ができる

$$\Downarrow$$

表示方法の検討

- 攻撃関係を辿らず非連結部分として表示 (現在のバージョン)
- 攻撃関係を辿りすべて表示
- 非連結部分は表示しない

攻撃関係を辿る意味があるか

- ボトムアップ推論の意味を考えると表示すべきでない
- 攻撃関係を辿らないとトップダウン推論で何度も無駄な処理をする
- すべてを表示すると煩雑な表示になる
- 何も表示しないとせっかく調べた証拠の意味がわからなくなる

トップダウン推論における OR 分岐の扱い

アルゴリズム中に生じる OR 分岐

- 差分支持集合が複数ある場合
- あるノードから下向きに辿る OR 関係経路が複数ある場合

扱い

- 手法としてはすべて実行可能
- 現在はすべての可能なものを表示してユーザが選択
- 候補の優先度をつけ，高いものを推奨する方法を検討中

候補の優先度の定義

トップダウン推論では必要な証拠を示唆するのが目的

$\Downarrow$

要求されるする証拠からの筋だてが強いものが有利

- 要求される証拠の数
- 反論の数
- 証拠に至るまでの自分への支持の長さ (?)

$\Downarrow$

これらを反映したノードや経路の評価値の設定

アウトライン

- 双極議論フレームワーク
- BAF 上の双方向推論システム
- 検討中の課題
- おわりに

結論と今後の課題

結論
BAF 上の法律推論システムの提案と実装

$\Downarrow$

- ユーザが裁判や法律推論をシミュレートすることで理解を助ける
- 議論一般への拡張の可能性

今後の課題

- 攻撃関係の扱い
- OR 分岐の扱いと分岐の優先付け
- 実際の法律への適用と評価