

妥当な推測を表すためのグラフモデルと それに基づく仮説構築支援システム

西野 正彬[†] 山本 章博[†] 林 晋^{††}

[†] 京都大学 大学院情報学研究所

^{††} 京都大学 大学院文学研究科

あらまし 本稿では、研究者の生成する仮説を表現するためのグラフモデルである Reasoning Web モデルを定式化する。Reasoning Web モデルは仮説中に存在する根拠と根拠からの帰結との間の依存関係をリンクとして表現するグラフモデルであり、このモデルに基づいて仮説を表現することで、不整合が生じることなく仮説を更新することが可能となる。Reasoning Web モデルの定式化とともに、本研究で開発した Reasoning Web モデルに基づく仮説構築支援システムについても述べる。利用者はシステムを利用することによって、対話的に仮説を構築することが可能である。
キーワード グラフモデル, 仮説の構築, 仮説の更新, 支援システム

Graphical Representation of Valid Conjectures and a System for Supporting Construction of Hypotheses

Masaaki NISHINO[†], Akihiro YAMAMOTO[†], and Susumu HAYASHI^{††}

[†] Graduate School of Informatics, Kyoto University

^{††} Graduate School of Letters, Kyoto University

Abstract In this paper, we formulate the *Reasoning Web model*, a new model for graphical representation of hypotheses that researchers create in their academic research. In the Reasoning Web model, a hypothesis is represented as a graph where a link is interpreted as the dependency between a consequence and its evidence. With the Reasoning Web model, we can update hypotheses without losing its validity by using the dependency in the graph. Moreover, we report on a system which supports the construction of hypotheses based on the Reasoning Web Model. Users of our system can construct their hypotheses interactively.

Key words graphical representation, construction of hypotheses, updating hypotheses, supporting system

1. はじめに

近年、Google Book Search や国立国会図書館デジタルアーカイブポータル (PORTA)^(注1) など、デジタルアーカイブ化された歴史的文献の WWW 上での公開が進んでいる。こういった動きが進むにつれ、紙媒体であったこれら文献をもとにして研究を行ってきた分野において、扱う対象がデジタル文献へと移行することが予想される。そして、研究者が自身の仮説をデジタル化された文献と結びつけて計算機上やネットワーク上で表現するという研究形態が有望となると考えられる。

特に人文科学系の分野では、自然科学系の分野における帰納的実験のように、仮説の妥当性を保証するような明確な方法が存在しないことがある。そのようなばあい、仮説がどのような

根拠に依存しているのかという仮説中の根拠とそこからの帰結の依存関係が仮説の妥当性を定める。本稿では、この仮説中の依存関係を表現するためのグラフモデルである **Reasoning Web モデル** (以下、RW モデルと書く) を提案する。

仮説の依存関係は、特に仮説に更新を加えるときに重要となる。RW モデルによって仮説を表現することにより、更新によって仮説に不整合が発生することを防ぐことができる。さらに仮説中に矛盾が発生したばあいには、利用者がそれを除去することを支援することも可能となる。本稿では RW モデルに基づく仮説の構築と更新を支援するシステムについても報告する。

なお、RW モデルでは仮説の表現に集合および集合間の演算を行う関数を利用するが、我々が実装する仮説構築システムでは集合・関数の記述方法を用意しない。これはある分野における仮説を表現するために十分な記述方法を設定することの困難さによる。そのかわりにこれらの集合および関数をテキストで

(注1) : <http://books.google.com/>, <http://porta.ndl.go.jp/>

表現し、利用者がテキストを解釈して必要な演算に対応する操作を行うことで仮説を構築するシステムとして実装する。つまり、関数や集合の実体はテキストを理解する利用者の思考の内部にあるとして実装する。詳しくは4章で述べる。

本研究は、小林[1]によるSMART-GSシステムの構成要素であるReasoning Webシステムを基礎としている。SMART-GSシステムは、デジタル画像中の手書き文字検索など、文献学研究のために有用な機能を備えている。SMART-GSシステムは、Hayashiら[2]によるシステム開発のためのUMLモデリングツールであるSMARTに基づいている。

SMART-GSのReasoning Webシステムは、文献解釈時の仮説をリンク構造として表現するためのシステムであり、変更影響分析機能とよばれる、ある部分の変更の際に、次にどの部分を変更すればよいかの指標を示す機能を備えている。本研究では、小林のReasoning Webシステムを基礎とし、リンクに対して根拠とその根拠からの帰結を結ぶという意味をもたせることで、リンク構造に対して与えられる変更の影響をより厳密に求めることができるようにした。

また、仮説を表現することが可能なグラフモデルとして、Truth Maintenance System[3]およびBayesian Network[4]といったものが挙げられる。これらはどちらも更新される仮説を表現することができるが、研究者の仮説を表現するという本研究の目的には沿わないと考える。

2. Reasoning Web モデル

Reasoning Web モデルは有限グラフ $G = \langle V, E \rangle$ によって仮説を表現する。 V はノードの有限集合、 E はリンクの有限集合である。まず、グラフの構成要素であるノードとリンクにどのような意味をもたせるかについて述べる。

推測の根拠とする事象、また、帰結とする事象をグラフモデルのノードとして扱う。各ノードにはそのノードが表現する対象に対応する名前が与えられ、それによって一意に識別可能であるとする。また、1つのノードのラベルは何らかの属性値をとることができるような変数であり、かつそれぞれのノードはそのノードが表現したい単一の概念と対応付けられるとする。それぞれのノードのラベルがとる属性値は、そのノードが対応付けられているものの性質を表現する。なお、以下ではノードとそのラベルを区別せずに u, v, w, \dots などと表記する。

ノードの属性値に対して、以下を定義する。

[定義 2.1] G 中のすべてのノード $v \in V$ に対し、 v の属性値が取り得る値全体の集合を仮想的に定め、それを v の定義域とし、 $\text{dom}(v)$ と表す。さらに、 v には $d(v) \in \text{dom}(v)$ である単一の要素 $d(v)$ が関連付けられているとする。 $d(v)$ を v の真の値とよぶ。

真の値は、 v の属性値を観測することが可能としたばあい観測される値であり、すべてのノードについて観測されるわけではないとする。真の値が観測可能かどうかは以下のように状態を用いて表現する。

[定義 2.2] すべてのノード $v \in V$ に対し、定義域 $\text{dom}(v)$ の部分集合を関連付ける。これを v の状態とし、 $\text{val}(v)$ と表す。

$\text{val}(v)$ は真の値 $d(v)$ が含まれると考えられている集合である。

ノード v の状態 $\text{val}(v)$ は $d(v)$ の性質に対応する集合と解釈可能である。 $\text{val}(v) = d(v)$ であるなら、それは $d(v)$ が観測可能であることを表す。一方、 $\text{val}(v) = \text{dom}(v)$ であるならば、 $d(v)$ が観測不可能であることを表す。そのいずれでもなく、 $\text{val}(v) \subset \text{dom}(v)$ ならば、それは $d(v)$ がとるある性質に対応する。

推測における根拠と根拠からの帰結の依存関係をリンクによって表現する。リンクは推測行為を表現するため、根拠からその推測の帰結への方向をもつ。リンクの導入にあわせ、以下を定義する。

[定義 2.3] $G = \langle V, E \rangle$ が、ループを含まないとき、Reasoning Web モデルによって表現された仮説とよぶ。

[定義 2.4] ある仮説に対し、その仮説の中であらかじめ $\text{val}(v) \neq \emptyset$ であるようなノード v を、その仮説の前提とよび、すべての前提の集合を Pr とする。 $Pr \subseteq V$ である。なお、すべての前提 $v \in Pr$ は、仮説中のいかなるリンクに対しても、そのターゲットとして出現することがないとする。

[記法 2.5] 各リンク $e \in E$ に対し、その始端に結合されているノード、すなわちリンクのソースノードの集合を $\text{src}(e)$ と表す。同様に終端に結合されているノード、すなわちターゲットノードを $\text{tgt}(e)$ と表す。

[定義 2.6] ノード $v \in V$ に対し、 $v \in \text{src}(e)$ となるようなリンク e の集合を $\text{slink}(v)$ と表す。同様に $v = \text{tgt}(e)$ となるリンクの集合を $\text{mlink}(v)$ と表す。

仮説においては、根拠と帰結間に依存関係が存在するということのみではなく、その依存関係がなぜ成立するかという推測理由が重要視される。そこで、RW モデルでは推測理由を以下のように根拠から帰結を得る関数であると定義し、リンク e に関連付ける。

[定義 2.7] 各ノード $e \in E$ に与えられる推測理由 r_e は、 $r_e : 2^{\text{dom}(v_{s1})} \times 2^{\text{dom}(v_{s2})} \times \dots \times 2^{\text{dom}(v_{sN})} \rightarrow 2^{\text{dom}(v_t)}$ である関数とする。ここで、 $\{v_{s1}, v_{s2}, \dots, v_{sN}\} = \text{src}(e)$ 、 $v_t = \text{tgt}(e)$ である。

推測理由 r_e を関数とすることで、あるノードの状態がそのノードをターゲットとするリンクによって定まる様子、すなわちある根拠をもとにした推測理由によって帰結が定まる様子が表現される。

[定義 2.8] 推測理由 r_e が、 $v_t = \text{tgt}(e)$ に与える帰結 c_e を

$$r_e(\text{val}(v_{s1}), \text{val}(v_{s2}), \dots, \text{val}(v_{sN})) = c_e$$

として定義する。なお、 $\{v_{s1}, v_{s2}, \dots, v_{sN}\} = \text{src}(v)$ である。

[定義 2.9] あるノード $v \in V$ に対し、 $e \in \text{mlink}(v)$ なるすべてのリンク e から v に与えられた帰結 c_e の集合を、 $\text{cnsq}(v)$ と表す。

[定義 2.10] 仮説の前提ではないあるノード $v \in V$ に対し、その状態 $\text{val}(v)$ を $\text{val}(v) = \bigcap_{c \in \text{cnsq}(v)} c \cap \text{dom}(v)$ とする。

ここで特別なリンク v_T を導入する。 v_T は、 $v \in Pr$ であるすべてのノード v についての仮想的な親ノードとして導入されるノードである。これにより、 v の状態は v_T からのリンク e

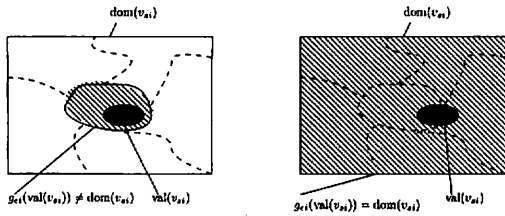


図1 性質関数 g_{ei} の概念図

の帰結によって与えられるとされる。こうして v_T を除くすべてのノードについて、その状態が定義 2.10 によって与えられるとして扱うことが可能となる。

r_e は推測理由を表現する関数であるとしたが、推測とは、「根拠がある性質を満たすことから、ある推測理由にもとづいて帰結を与える」行為である。つまり、推測は (i) 根拠がある性質を満たすことを確かめ、(ii) それに基づいて帰結を与える、という 2 つの段階に分けることができる。そこで、推測理由をそれぞれの段階に対応する関数の組み合わせとして、以下のように表す。

[定義 2.11] リンク $e \in E$ に関連付けられた推測関数 r_e を関数 g_{ei} と h_e の合成として表現する。つまり、

$$r_e(\text{val}(v_{s1}), \text{val}(v_{s2}), \dots, \text{val}(v_{sN})) = h_e(g_{e1}(\text{val}(v_{s1})), \text{val}(v_{s2}), \dots, g_{eN}(\text{val}(v_{sN})))$$

とする。関数 g_{ei} ($1 \leq i \leq N$) を、それぞれソースノード v_{si} に割り当てられた性質関数、 h_e を推測関数とよぶ。

性質関数、推測関数は以下のように定義される。

[定義 2.12] リンク $e \in E$ の i 個目のソースノード $v_{si} \in \text{src}(e)$ に割り当てられる推測関数 g_{ei} は、以下の性質を満たすものとする。

- (1) $g_{ei} : 2^{\text{dom}(v_{si})} \rightarrow 2^{\text{dom}(v_{si})}$.
- (2) $X \in 2^{\text{dom}(v_{si})}$ なる X に対し、 $g_{ei}(X) \supseteq X$.
- (3) $X \in 2^{\text{dom}(v_{si})}$ なる X に対し、 $g_{ei}(X) = g_{ei}(g_{ei}(X))$.
- (4) $g_{ei}(\emptyset) = \emptyset$.
- (5) $Y \in 2^{\text{dom}(v_{si})}$ に対し、 $g_{si}(X) \cap Y \neq \emptyset$ かつ $g_{si}(X) \not\subseteq Y$ であるような $X \in 2^{\text{dom}(v_{si})}$ が存在するならば、 $g_{si}(Y) = \text{dom}(v_{si})$.

定義から得られる性質関数の概観を図 1 に示す。直感的には、 g_{ei} はノード v_{si} の定義域 $\text{dom}(v_{si})$ を分割し、現在の状態 $\text{val}(v_{si})$ がどの分割に含まれるかを判断する関数である。もし、いずれかの分割に含まれているならば、その分割に等しい集合を返し、一つの分割に含まれていないならば、 $\text{dom}(v_{si})$ に等しい集合を返す。それぞれの集合が推測関数 h_e が表す推測において根拠とできる性質を示し、もし $g_{si}(\text{val}(v_{si})) = \text{dom}(v_{si})$ であったならば、推測の根拠として適切でないことを表す。

次に推測関数の性質を示す。

[定義 2.13] リンク $e \in E$ に対応する推測関数 h_e は以下の性質をもつ。

- (1) $h_e : 2^{\text{dom}(v_{s1})} \times \dots \times 2^{\text{dom}(v_{sN})} \rightarrow 2^{\text{dom}(v_t)}$ 。ただし、 $\{v_{s1}, v_{s2}, \dots, v_{sN}\} = \text{src}(e)$ 、 $v_t = \text{tgt}(e)$ である。

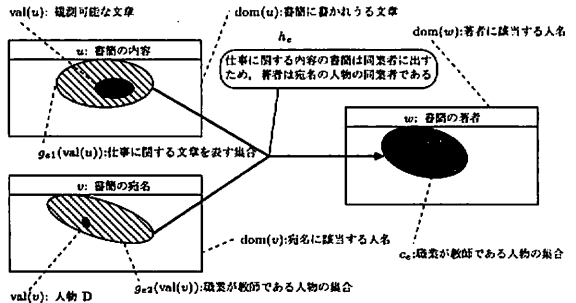


図 2 Reasoning Web モデルによって表現された仮説の例

(2) $g_{ei}(\text{val}(v_{si})) = \text{dom}(v_{si})$ となるような性質関数 g_{ei} ($1 \leq i \leq N$) が一つでも存在するならば、 $h_e(g_{e1}(\text{val}(v_{s1})), \dots, g_{eN}(\text{val}(v_{sN}))) = \text{dom}(v_t)$.

定義の 2 つめの条件は、いずれかの性質関数において $g_{ei}(\text{val}(v_{si})) = \text{dom}(v_{si})$ であったばあい、すなわち v_{si} の状態が推測の根拠として適切でなかったばあいに、 h_e が v_t に対する有効な帰結を与えないことを意味する。

[定義 2.14] リンク $e \in E$ のソースノードを $v_{s1}, v_{s2}, \dots, v_{sN} \in \text{src}(e)$ とし、それぞれのノードに割り当てられた性質関数を $g_{e1}, g_{e2}, \dots, g_{eN}$ とする。 $g_{ei}(\text{val}(v_{si})) = \text{dom}(v_{si})$ であることによって、 h_e の出力が $\text{dom}(v_t)$ となっているとき、 h_e が g_{ei} によって無効化されているとよぶ。同様に、リンク e を無効化されたリンクとよぶ。その反対に、無効化されていないリンクを有効なリンクとよぶことにする。

[例 2.15] ある著者不明の書籍に対し、既知であるその宛名と内容より「仕事に関する内容の書籍は同業者に宛てるものであるから、著者は宛名の人物 D の同業者である」という推測理由に基づいて著者を推測する仮説は、「内容」ノード u 、「宛名」ノード v 、「著者」ノード w によって図 2 のように表現される。

3. 仮説の更新

研究者の生成する仮説は更新される可能性がある。RW モデルによって表現された仮説にとつて、更新とはその構造を変化させることに相当する。

仮説の更新においては仮説の依存関係を考慮する必要がある。現在の仮説のある一部分を更新したい、それに依存する部分も同時に更新しなければ、仮説に不整合が生じる可能性がある。

RW モデルで表現された仮説は、その依存関係を把握できるため、そのような不整合を生じない更新を行うことができる。とくに仮説中に矛盾が発生したばあいには、矛盾原因除去手続きによってその矛盾の原因を除去することができる。

3.1 仮説の矛盾

仮説は誤りを含む可能性がある。RW モデルは表現された仮説の正しさを判断する仕組みをもたないものの、矛盾を検出することによって仮説に誤りが生じていることを検出することができる。矛盾が発生している以上は仮説が妥当であるとはいえないため、RW モデルでは矛盾を除去する手続きを実行する。

まず、以下のように矛盾を定義する。

[定義 3.1] あるノード $v \in V$ の状態が $\text{val}(v) = \emptyset$ のとき、 v で矛盾が発生しているとはよぶ。

RW モデルでは v_T を除くすべてのノード $v \in V$ の状態は $e \in \text{tlink}(e)$ に付随する推測理由 r_e にもとづいて定まるとしていた。そのため矛盾の発生はいずれかのリンクに関連付けられた推測理由に誤りがあることを意味する。矛盾を除去するためにはそのような推測理由を与えているリンクを除去すればよい。

RW モデルは推測理由の正しさを判断できないため、矛盾除去のために以下の手順をとる。まず、除去することによって矛盾を除去できるようなリンクの集合を矛盾原因候補であるととし、そのようなリンクの集合をすべて列挙する。その後、列挙された矛盾原因候補を外部システムに入力として与える。外部システムはリンクの推測理由を理解し、列挙された矛盾原因候補から矛盾原因であると考えられる候補の一つ選ぶことができる。こうして外部システムから選ばれた矛盾原因候補に含まれるリンクを除去することによって矛盾を除去できる。

矛盾原因候補を列挙するためには、どうすれば現在の矛盾が除去できるかを知る必要がある。いま、矛盾が発生しているノードの集合を $V_0 \subseteq V$ とする。矛盾を除去する、すなわち $v \in V_0$ なるすべての v に対して $\text{val}(v) \neq \emptyset$ とするためには、各 v において $c_e \in \text{cnsq}(v)$ である帰結 c_e をいくつか除去あるいは無効化すればよい。なお、以降では帰結 c_e を除去あるいは無効化することを、その帰結を取り下げるとよぶ。

矛盾原因候補列挙の説明にはいる前に、以下を定義する。

[定義 3.2] ノード $v \in V$ の定義域の部分集合 $X \in 2^{\text{dom}(v)}$ に対し、 $X \not\subseteq \bigcap_{c \in C} c$ であるようなすべての $C \in 2^{\text{cnsq}(v)}$ からなる集合を、 X の非包含帰結集合とよび、 $NS(X)$ と表す。 $C \in NS(X)$ なる C のうち、 $C \subset C'$ であるような $C' \in NS(X)$ が存在しないものからなる集合を極大非包含帰結集合とよび、 $MNS(X)$ と表す。

[定義 3.3] ノード $v \in V$ の定義域の部分集合 $X \in 2^{\text{dom}(v)}$ に対し、 $X \supseteq \bigcap_{c \in C} c$ の条件を満たす $C \in 2^{\text{cnsq}(v)}$ からなる集合を、支持帰結集合とよび、 $SC(X)$ と表す。さらに、 $C \in SC(X)$ のうち、 $C \supset C'$ であるような $C' \in SC(X)$ が存在しないもの集合を極小支持帰結集合とよび、 $MSC(X)$ と表す。

特に、 $X = \emptyset$ のときの極小支持帰結集合を極小矛盾原因帰結集合とよぶことにする。

帰結の組み合わせ $C \in MSC(X)$ は、直感的には $\text{val}(v)$ が集合 X によって表現される性質をもつ要因であるとして解釈できる。

なお、[5] より、集合 $X \in \text{dom}(v)$ に対する極小支持帰結集合 $MSC(X)$ の極小ヒッティングセットは $\overline{MNC}(X)$ であることが証明できる。 $\overline{MNC}(X)$ は $C \in MNC(X)$ なる帰結の集合 C の $\text{cnsq}(v)$ に対する補集合からなる集合である。

矛盾原因候補であるリンクの集合を列挙するために、まず各リンク $e \in E$ に対応する命題変数によって論理式を構成する。そして、その論理式を真とするような命題変数の組み合わせが矛盾原因候補であるとして、そのような組み合わせを列挙する。

各リンク $e \in E$ に対応する命題変数による論理式として、以下を定義する。なお、リンク e に対応する命題変数は e と表現

する。

[定義 3.4] 矛盾が発生しているノードの集合を $V_0 \subseteq V$ とし、 V_0 に論理式 $\text{Rc}(V_0)$ を関連付ける。 $\text{Rc}(V_0)$ が真となるときに、 $v \in V_0$ での矛盾は除去されるとする。

[定義 3.5] 無効化されていないリンク $e \in E$ に関連づけられた帰結 c_e に対し、論理式 $\text{Inv}(c_e)$ を関連付ける。 $\text{Inv}(c_e)$ が真となるときに c_e は取り下げられるとする。

[定義 3.6] ノード $v \in V$ の定義域の部分集合 $X \in 2^{\text{dom}(v)}$ に対し論理式 $\text{Nin}(X)$ を関連付ける。 $\text{Nin}(X)$ が真となるときに、 $X \not\subseteq \text{val}(v)$ であるとする。

矛盾原因候補を列挙するためには、 $\text{Rc}(V_0)$ を真とするような変数の組み合わせを求めればよい。その手順は以下のようになる。

$$\begin{aligned} \text{Rc}(V_0) &= \bigwedge_{v \in V_0} \text{Nin}(\text{val}(v)) \\ \text{Nin}(X) &= \bigvee_{C \in \overline{MNS}(X)} \bigwedge_{c \in C} \text{Inv}(c) \\ \text{Inv}(c_e) &= \bigvee_{v_{\text{src}} \in \text{src}(e)} (\text{Nin}(g_{\text{est}}(v_{\text{src}}))) \vee e \end{aligned}$$

$\text{Rc}(V_0)$ は、すべての $v \in V_0$ について $\text{val}(v) \neq \emptyset$ となるときに真であるとしている。 $\text{Nin}(X)$ では、 $X \not\subseteq \text{val}(v)$ となる条件が、いずれかの $C \in \overline{MNS}(X)$ に含まれるすべての帰結 $c \in C$ を取り下げることでありとしている。そして $\text{Inv}(c)$ では、帰結を取り下げるためにはその帰結が与えられるリンク e を除去するか、またはそのいずれかのソースノード $v_{\text{src}} \in \text{src}(e)$ において、 $\text{Nin}(g_{\text{est}}(\text{val}(v_{\text{src}})))$ とすることで無効化する必要があることを示している。これらの式を再帰的に組み合わせることで最終的に $\text{Rc}(V_0)$ がリンクに対応する命題変数のみからなる論理式として得られるため、これを満たすリンクの組み合わせを矛盾原因候補として列挙する。

$\text{Rc}(V_0)$ は正リテラルのみからなる論理式のため、それを DNF に変換すればその各項が矛盾原因候補であるリンクの集合となっている。よって各項のうち、極小であるものを矛盾原因候補であるとして列挙することにする。こうして $\text{Rc}(V_0)$ を求めることから矛盾原因候補の列挙が可能となる。

3.2 スコアによる仮説の評価

RW モデルは推測理由を知ることができないとしたが、そうであったとしても何らかの近似を与えるスコアを導入することは有用である。そのようなスコアとして信頼度と影響度を導入する。信頼度は外部からモデルに対して与えられるスコアで、あるリンクの帰結やノードの状態がどれだけ信じられるかを反映する。一方、影響度は仮説の構造のみから定まるスコアで、あるリンクの仮説における影響の大きさを反映する。

3.2.1 信頼度

信頼度はノードの状態、リンクおよびリンクの帰結に対して定義されるスコアである。0~1 の実数で与えられ、そのスコアが大きいほど確からしいことを表現する。リンクに対するスコアは、外部よりその推測理由の確かさの指標として与えられ、それに基づいてノードの状態、リンクの帰結のスコアが定まる。

リンク、リンクの帰結、そしてノードの状態に対する信頼度をそれぞれ $\text{conf}^E, \text{conf}^C, \text{conf}^V$ として以下のように定義する。
 [定義 3.7] リンク $e \in E$ に対する信頼度を $\text{conf}^E(e)$ とする。
 $\text{conf}^E(e)$ は e が追加されるさいに外部から与えられるとする。
 [定義 3.8] リンク $e \in E$ に付随する帰結 c_e に対する信頼度を

$$\text{conf}^C(c_e) = \min_{(1 \leq i \leq N)} (\text{conf}^V(g_{ei}(\text{val}(v_{si})))) \cdot \text{conf}^E(e)$$

とする。 $\{v_{s1}, \dots, v_{sN}\} = \text{src}(e)$ であり、 g_{ei} は v_{si} に割り当てられた性質関数である。

[定義 3.9] ノードに対する信頼度は、そのノードの定義域の部分集合に対して与えられる。すなわち、あるノード $v \in V$ の定義域の部分集合 $X \in 2^{\text{dom}(v)}$ に対し、信頼度 $\text{conf}^V(X)$ を

$$\text{conf}^V(X) = \max_{C \in \text{MSC}(X)} (\min_{c \in C} (\text{conf}^C(c)), 0)$$

とする。 $\text{MSC}(X)$ は X に対する極小支持帰結集合である。

3.2.2 影響度

信頼度は外部からリンクに対して与えられるため、そのスコアの与え方に依存する。一方、影響度は純粋にリンク構造のみを反映するスコアとして定義する。

影響度は矛盾原因除去手続きで利用されることを想定し、リンクの集合に対するスコアとして以下のように定義する。

[定義 3.10] リンクの集合 $X \in 2^E$ に対し $e \in X$ なるすべてのリンク e を除去したさいに、除去あるいは無効化されるリンクの総数をその影響度とし、 $\text{imp}(X)$ とする。

影響度を利用することで、あるリンクが仮説中のどのリンクによって参照されているかを反映することができる。

3.3 仮説更新手続き

矛盾除去、スコアを考慮したうえで、RW モデルにおける仮説の更新手続きについて述べる。仮説の更新はリンク構造の更新であるから、その操作はノードの追加・変更・削除およびリンクの追加・変更・削除の 6 種類の操作に限定される。

これらの操作のうち、ノード・リンクの変更は、対象であるノードまたはリンクを削除したのちに新たなノードまたはリンクを追加することで達成できる。また、新しいノード v を追加したとしても、 v が他のノード $u \in V$ とリンクで結ばれていない限りは他のリンクおよびノードに影響を与えることはない。ノードの削除については、除去するノードをソースあるいはターゲットとするリンクをすべて除去すれば同等の結果を得ることが可能である。よって、リンクの追加・削除に対してのみ、その影響による更新を行えばよい。

RW モデルにおいて、更新の対象となるのは各リンク $e \in E$ の帰結 c_e およびその信頼度 $\text{conf}^C(c_e)$ 、そして各ノード $v \in V$ の状態 $\text{val}(v)$ であり、それらはすべてその祖先であるリンクの帰結およびノードの状態より定められていた。これをふまえ、仮説の更新手続きは図 3 の update のようになる。

この仮説更新手続きでは、ノードおよびリンクにマークを付加することによって、常に祖先から子孫の順で更新が行われるようにしている。このように更新を行うことにより、一旦更新されたノードおよびリンクにおいて、現在の更新が停止するま

$\text{update}(X)$

```

1: global conts ← ∅ /* 矛盾が生じたノードの集合 */
2: for all e ∈ X do
3:   e の子孫であるすべてのリンク・ノードに
   NEED.TO.UPDATE マークを付ける
4: for all e ∈ X do
5:   update_node(tgt(e))
6: if conts ≠ ∅ then /* 矛盾が発生 */
7:   Y ← 矛盾原因除去手続きによって指定された矛盾原因
8:   for all e ∈ Y do
9:     remove_link(e) /* e を仮説から除去 */
10:  update(Y) /* 矛盾除去による更新 */
  
```

$\text{update_node}(v)$

```

1: for all e ∈ tlink(v) do
2:   if mark(e) = NEED.TO.UPDATE then
3:     update_link(e)
4: if mark(v) ≠ NEED.TO.UPDATE then
5:   return
6: if val(v) = ∅ と判定された then
7:   conts ← {v} ∪ conts
8:   v と v の子孫であるリンク・ノードに CONTRA-
   DICTED マークを付ける。
9: else /* 矛盾と判定されなかった */
10:  mark(v) ← UPDATED
11:  for all e ∈ slink(v) do
12:    update_link(e)
  
```

$\text{update_link}(e)$

```

1: for all v ∈ src(e) do
2:   if mark(v) = NEED.TO.UPDATE then
3:     update_node(v)
4: if mark(e) ≠ NEED.TO.UPDATE then
5:   return
6: c_e ← h_e(g_{e1}(val(v_{s1})), g_{e2}(val(v_{s2})), ..., g_{eN}(val(v_{sN})))
7: conf^C(c_e) ← 定義 3.8 にもとづいて計算する。
8: mark(e) ← UPDATED
9: update_node(tgt(e))
  
```

図 3 仮説更新アルゴリズム

でに再び更新が発生することがなくなる。

更新において矛盾が発生したばあい、矛盾が発生したノードの子孫の更新をいったん保留し、更新可能な部分の更新がすべて完了したのちに、矛盾除去手続きを実行する (update : ステップ 6 以降)。その後、矛盾原因候補の除去に対応する更新を行う。

4. 仮説構築支援システム

RW モデルに基づいた仮説構築支援システムを実装する。第 2 章でノードの状態が集合として与えられること、推測理由が関数として与えられることを述べてきたが、実際にシステムと

して実装するためには、

- 各リンクの帰結の集合による表現。
- 推測理由を表す関数の手続きを記述する記述法。

を導入しなければならない。冒頭でも述べたように、本実装ではこれらをすべて利用者が理解できるテキストとして与え、必要な場面でダイアログを通じて利用者が判断を行うシステムとして実装する。

さらに、3.1節で述べた矛盾除去における外部システムに対応する処理、すなわち矛盾原因候補から一つを選択する行為も利用者が行うことにする。そのさいに信頼度・影響度を利用して列挙の方法を工夫する。

4.1 利用者の介入の必要性

外部システムに相当する判断以外に、利用者が判断および実行しなければならないのは、

- 矛盾の発生および矛盾原因の特定。
- 性質関数の極小支持帰結集合。
- 根拠と推測理由から帰結を獲得する行為。

の3点である。

矛盾の発生およびその矛盾原因の特定については、あるノード $v \in V$ に対して新しい帰結が追加されるか、もしくは既存の帰結が変更されるたびに、そのノードにおいて矛盾が発生していないかを利用者が判断すればよい。各帰結はテキストで表現され、その箇条書きが表す性質をすべて満たすものとして状態が表現されるため、矛盾の判断は、その箇条書きに矛盾がないかを判断すればよいことになる。

矛盾が発生していたばあいには、矛盾除去手続きを開始するために、利用者に極小矛盾原因帰結集合を指定させる。箇条書きから矛盾を判断できるなら、極小矛盾原因帰結集合を指定することも容易であると考ええる。

性質関数の極小支持帰結集合を指定することができれば、そこから極大非包含帰結集合も得ることができるため、矛盾発生時の矛盾原因候補の列挙や信頼度の計算およびリンクの無効化の判定をシステム側で行うことができる。そこで、利用者にはリンクを追加するさい、およびリンクのソースノードに新しい帰結が追加されたさいに、その性質関数に対する極小支持帰結集合を入力するよう求める。極小支持帰結集合の指定は、利用者がある推測において何を根拠としたかを指定すればよいので直感的に指定できると考える。

根拠と推測理由から帰結を獲得する行為、つまり推測理由に対応する関数の実体に相当する演算は、実際に根拠の状態と推測理由を確認しながら利用者が行うことになる。仮説が更新されたさいには、その更新を反映するように演算をやり直す必要があるが、そのような演算が必要となる条件は、各ソースノードに与えられた帰結の更新状況に依存するため、システムは必要となしにのみ利用者に判断を求めることにする。

矛盾除去については、列挙するさいにスコアを利用して矛盾原因候補の列挙の順序を変化させる。利用するスコアは、矛盾原因候補であるリンクの集合の信頼度および影響度、そしてその候補を選択することで取り下げられる、矛盾が発生しているノードに与えられる帰結の信頼度を利用する。

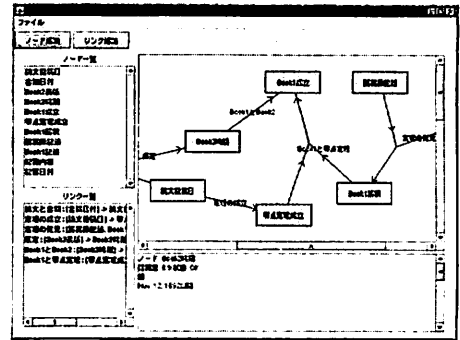


図4 システムの概観

4.2 システムの実装

我々が実装したシステムの概観を図4に示す。実装にはJavaを用いた。利用者はGUIを通じてRWモデルによる仮説の構築を行うことができる。更新時には、どのような更新が行われたかを把握することにより、利用者の判断が必要となる条件を特定することができる。そのようにして判断された条件に合致するときには、逐次ダイアログを提示することで利用者に判断を求めようとした。

5. おわりに

本稿では、研究者の仮説を表現するためのグラフモデルであるReasoning Webモデルの設計と、そのモデルに基づいた仮説構築支援システムの開発について述べた。

Reasoning Webモデルは、妥当性のある推測における帰結とその根拠の間に依存関係があることに着目し、その依存関係をリンクとして構成するグラフとして仮説を表現する。このように仮説の依存関係を表現することによって、その更新時に不整合が発生することを防ぐことができる。また、仮説に矛盾が生じたばあいには矛盾原因候補であるリンクの集合を列挙し、そこから一つ候補を選択することで、矛盾を除去できる。

本研究ではReasoning Webモデルを研究者が自身の仮説を表現するためのモデルであるとして構築したが、今後他の研究者との協調作業を表現するための拡張を行うことは有用だと考える。冒頭で述べたような文献デジタルアーカイブ化にあわせて、WWW上でこのような協調作業を行うことを可能とするシステムを構築できれば、その価値は高いと考える。

文 献

- [1] 小林: “文献学研究用ワークベンチの開発”, 京都大学大学院文学研究科 修士論文 (2006).
- [2] S. Hayashi, Y. Pan, M. Sato, K. Mori, S. Sul and S. Haruna: “Test driven development of UML models with SMART modeling system”, UML 2004, Vol. 3273 of LNCS, Springer, pp. 395-409 (2004).
- [3] J. Doyle: “A Truth Maintenance System”, Artificial Intelligence, 12, 3, pp. 231-272 (1979).
- [4] R. Neapolitan: “Learning Bayesian networks”, Pearson Prentice Hall Upper Saddle River, NJ (2004).
- [5] K. Satoh and T. Uno: “Enumerating maximal frequent sets using irredundant dualization”, DS 2003, Vol. 2843 of LNCS, Springer, pp. 256-268 (2003).