知能と複雑系 110－2
(1998. 1. 22)

タブロー法に基づく文脈推論のための決定手続き

小布施秀幸* 太原育夫**

東京理科大学 理工学部 情報科学科
千葉県野田市山崎2641
*0471-24-1501 (ext. 3356) obuse@spl.is.noda.sut.ac.jp
**0471-24-1501 (ext. 3311) tahara@is.noda.sut.ac.jp

あらまし McCarthy は「文脈 c で命題 p が真である」という言明を ist(c, p)という様相で表現し，文脈推論(contextual reasoning)の形式化を提案した. 本論文では，命題論理を対象にした文脈推論の形式化についてその意味論を検討し，文脈推論システムにおける質疑応答のための決定手続きとして prefix 付タブロー証明を検討した. そして，文脈推論において重要な役割を果たす lifting スキーマが prefix 付タブロー証明の中に無理なく組み込めることをいくつかの例によって示した.

キーワード 文脈，文脈推論，lifting スキーマ，prefix 付タブロー

Tableau-based decision procedure for contextual reasoning

Hideyuki Obuse* Ikuo Tahara**

Dept. of information sciences, Faculty of Science and Technology
Science University of Tokyo
2641 Yamazaki, Noda, Chiba 278, Japan
*0471-24-1501 (ext. 3356) obuse@spl.is.noda.sut.ac.jp
**0471-24-1501 (ext. 3311) tahara@is.noda.sut.ac.jp

Abstract McCarthy formalized contextual reasoning by introducing the ist(c, p) modality, which means that p is true in the context c. In this paper, we consider the propositional logic of context, and discuss the semantics. Prefixed tableau system is proposed as a decision procedure for the query-answering in contextual reasoning. Examples show that the lifting schema, which plays important role in contextual, can be naturally incorporated into the process of prefixed tableaux.

key words context, contextual reasoning, lifting schema, prefixed tableaux

1. はじめに

我々は日常生活において、不完全な情報のもとに何らかの決定をしなければならないことがしばしばある。こうした場合我々は、常識を用いて不完全な部分を補い決定を行っていると考えられる。このような常識を利用した情報処理を形式化し、不完全な知識ベースのもとでの推論を可能にする試みとして、これまでデフォルト論理やサーカムスクリプションといった常識推論の枠組みが提唱されてきた[1]。

また、情報の不完全性のうち多義性に着目すると、従来より人間の情報処理においては状況や文脈の果たす役割の重要性が指摘されてきた。そして状況理論やそれをもとにした状況推論が提案されている[2][3]。

状況理論の特徴の一つは、表現や式の持つ意味がそれ単独では決まらず、状況を参照して初めて決まるという点にある。同じ表現でも、外部の状況が異なればその意味も変化する。さらに、推論に用いる規則自身が状況によって変わる推論も存在する。例えば、時速１２０キロという表現はプロ野球を見ている状況と車を運転している状況では意味が異なる。球の速度の意味をとるか、車の速度の意味をとるかその状況に依存している。もしプロ野球の状況で投手が投げた球の速度であれば、一般的に遅いということが推論されるし、車を一般道で運転している状況なら速いということが推論される。このように状況が異なれば表現や式の意味が異なり、まったく反対の式や表現が導かれることになる。

本論文では Guha や McCarthy が提案した文脈推論(contextual reasoning)の形式化をとりあげ、その意味論と決定手続きについて検討する。ここで、文脈推論とは状況あるいは文脈(以後 context と呼ぶ)を考慮した知識表現における推論のことであり、その特徴は「文脈 c で命題 p が真である」という言明を ist(c,p)という様相で表現し形式化しようという点にある。2.においてまず、context の命題論理についてその形式化の概略を述べ、3.においてその意味論について検討する。そして 4.において文脈推論システムにおける質疑応答のための決定手続きとしてタブロー法に基づく方法について検討する。最後にいくつかの例を示す。

2. context の形式化 [4]

2.1 context の命題論理

context を c、命題論理式を φ としたとき「c において φ が真である」ことを

$$ist(c,\varphi)$$

という文によって表現する。これはまた

$$c:\varphi$$

のようにも表現できる。前者は context c の外側から context c における φ をみた場合の表現で、後者は context c の内側で φ を見た場合の表現である。

そこでさらに

$$ist(c',ist(c,\varphi))$$

という表現も可能であり、これは

$$c':ist(c,\varphi)$$

ということになる。このように考えると ist は無限の再帰となり得るので最も外側の context を暗黙のうちに定め c_0 とすることにする。

このようにここでは命題論理を対象とし、文脈命題論理式を次のように定義する。

定義 1 p を命題、c を context としたとき、文脈命題論理式 φ、ψ は

$$\varphi,\psi ::= p \mid \neg\varphi \mid \varphi\wedge\psi \mid ist(c,\varphi)$$

ただし、その他の論理記号については

$$\varphi\vee\psi = \neg(\neg\varphi\wedge\neg\psi)$$
$$\varphi\to\psi = \neg\varphi\vee\psi$$
$$\varphi\leftrightarrow\psi = (\varphi\to\psi)\wedge(\psi\to\varphi)$$

とする。

2.2 Contet における推論

context c において推論を行う場合 context c の中に入ることができる。これを entering とよび、この結果 c: p となる。逆に context からでることも出来る。これを exiting と呼ぶ。context c に入った状態で p から q を推論し、context c から exiting すると ist(c, q)が得られることになる。

ある context において真である文の集合は別の context において真であるとは限らない。なぜなら他の context では異なる関係を持つかもしれないからである。context を導入する意味もここにあるといえる。

ある context の中で真であるものを別の context のなかでも真であると関連付けることを lifting という。lifting は次のようなスキーマで表現することができる。

$$c_0 : ist(c,\varphi) \rightarrow ist(c',\varphi)$$

3. context の意味論

context は命題の真偽を決定する枠組みとして考えることができる。そこで文脈命題論理式の解釈として下のように定義する。

定義 2 context c における解釈の集合を $M(c)$ とし、$m \in M(c)$ とする。このとき $c,m \models \varphi$ を以下のように定義する。

$$c,m \models p \quad iff \quad m \models p$$
$$c,m \models \varphi \wedge \psi \quad iff \quad m \models \varphi \text{ かつ } m \models \psi$$
$$c,m \models \neg\varphi \quad iff \quad m \not\models \varphi$$
$$c,m \models ist(c',\varphi)$$
$$iff\ \exists m' \in M(c') \text{について } c',m' \models \varphi$$

この定義は、文脈命題論理式 φ の解釈は古典命題論理における φ の解釈と異なり φ に対して多重の解釈ができるということを示している。すなわちある context では φ を真と解釈し別の context では偽と解釈することが可能だということである。さらに 1 つの context には 1 つの解釈が対応するという考え方をすれば定義 2 により簡単に以下のようにすることができる。

定義 3 context c に対応する解釈を $m(c)$ とする。このとき $c \models \varphi$ を以下のように定義する。

$$c \models p \quad iff \quad m(c) \models p$$
$$c \models \varphi \wedge \psi \quad iff \quad m(c) \models \varphi \text{ かつ } m(c) \models \psi$$
$$c \models \neg\varphi \quad iff \quad m(c) \not\models \varphi$$
$$c \models ist(c',\varphi) \quad iff \quad m(c') \models \varphi$$

定義 4 φ を文脈命題論理式、c を context とし

$$c,m \models \varphi$$

なる $m \in M(c)$ が存在するとき φ は c において充足可能であるという。

4. タブロー法を用いた決定手続き

4.1 prefix 付タブロー規則

文脈命題論理式の集合として知識 Σ が与えられたとき、Σ のもとである文脈命題論理式 φ が成り立つかどうかを決定する手続きを考えよう。

このことは意味論的には Σ を充足するすべての解釈において φ が真となるかどうかを調べれば良いので、ここでは $\Sigma \cup \{\neg\varphi\}$ の充足不能性をタブロー計算によって調べるという反駁的証明を考えることにする。タブロー規則を考えるにあたり、まず context 系列という表記法を導入する。すなわち一般に context は入れ子構造になるのでそれを明示して系列で表現することにする。たとえば、

$$c_0 : ist(c_1, bird)$$
$$c_1 : ist(c_2, fly)$$

のとき c_2 を $c_0c_1c_2$ と表記する。 したがって c_0 から c_1、c_1 から c_2 へ entering すれば

$$c_0c_1c_2 : fly$$

となる。(すなわち $c_0c_1c_2$ は c_2 と同じ context を示す別表現と考えることができる。)

さて、context 系列 $\bar{c}$ における $M(\bar{c})$ のうち i 番目の解釈を $\bar{c}[i]$ とすれば、様相論理における prefix 付タブロー法[5]と同様に考えて定義 2 より、図 1 に示すような prefix 付タブロー規則が得られる。

なお定義 3 に従えば例えば

$$\alpha : \frac{c[i] : \varphi \wedge \psi}{\begin{array}{c} c[i] : \varphi \\ c[i] : \psi \end{array}} \qquad \beta : \frac{c[i] : \neg(\varphi \wedge \psi)}{c[i] : \neg\varphi \mid c[i] : \neg\psi}$$

$$dneg : \frac{c[i] : \neg\neg\varphi}{c[i] : \varphi}$$

$$Pos - entering : \frac{\vec{c}[i] : ist(c, \varphi)}{\vec{c} \circ c[j] : \varphi}$$

ただし$\vec{c} \circ c[j]$は未出のもの

$$Neg - entering : \frac{\vec{c}[i] : \neg ist(c, \varphi)}{\vec{c} \circ c[j] : \neg\varphi}$$

ただし$\vec{c} \circ c[j]$は既出のもの

図1　タブロー規則

$$\frac{\vec{c} : ist(c, \varphi)}{\vec{c} \circ c : \varphi}$$

のように簡単なprefix付タブロー規則となる。

4.2 prefix付タブロー木

prefix 付タブロー木は節点に置かれた式に適用可能な prefix 付タブロー規則を適用していき拡張した二分木である。

定義5　タブロー木の枝に$\vec{c}[k] : \varphi$ と $\vec{c}[k] : \neg\varphi$ がともにある時その枝は閉じているという。適用可能なすべてのタブロー規則が適用されているときその枝は完全であるという。完全であって閉じていないとき開いているという。すべての枝が閉じているときもタブロー木は閉じているといい、少なくとも1つの枝が開いているときタブロー木は開いているという。

ここで lifting スキーマを考慮に入れる。上記の定義に加えて、lifting スキーマで指定された context c と c' についてタブロー木の枝に$c : \varphi$ と $c' : \neg\varphi$ がともにあればやはり枝は閉じているということにする。

定義5より、文脈命題論理式φ について $\vec{c}[0] : \neg\varphi$ から始まる prefix 付タブロー木が閉じているときφ は context 系列$\vec{c}$ によって充足不能であることがわかる。

prefix 付タブロー法による決定手続きは図 2 のようになる。

- **すべての前提および質問の否定をリストアップ**
- **繰り返し**
  - **前提または質問の否定にタブロー規則を適用**
  - **同じcontextまたはlifting関係にあるcontextが含まれるcontext系列に矛盾が存在すれば、その枝は閉じている**

if　すべての枝が閉じている
$\rightarrow$終了　質問は成り立つ
elseif　すべての前提と質問の否定にタブロー規則の適用が終了し、枝が閉じていない
$\rightarrow$終了　質問は成り立たない
else　繰り返し

図2　タブロー法による決定手続き

5 適用例

prefix 付タブロー法の適用例を二つあげる。例1は既存の非単調推論の枠組みでは解決不可能な YSP に対する適用例であり、例2は例外の処理に対する適用例である。

5.1 例1 (Yale Shooting Problem)

YSP[6]は、状態の変化に関しての情報と動作系列が与えられたときに、その動作系列上で何が成り立っているかを予測する問題である。

状態の変化の情報は以下のようなものである。

① 初期状況である人が生きている（Alive）
② 銃に弾丸を込める（Load）という動作が起こると銃に弾丸が込められた状態（Loaded）になる。
③ 銃に弾丸が込められた状態（Loaded）で銃を撃つ（Shoot）とその人は死ぬ（¬Alive）。

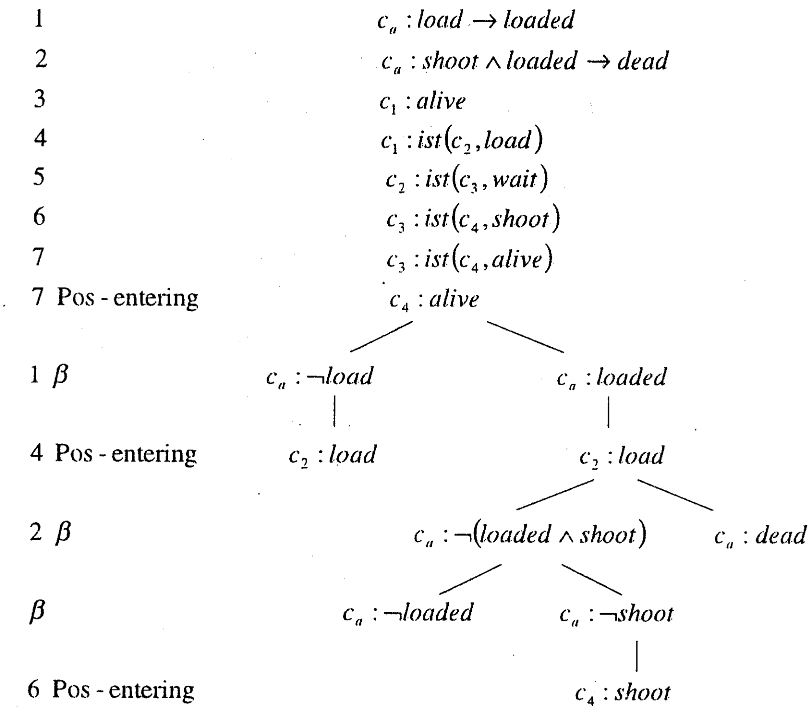

図3　タブロー木(例1)

動作系列としては、初期状況から銃に弾丸を込め（Load）、少し待って（Wait）、銃を撃つ（Shoot）というものが与えられたとする。

この時、各動作後の状況で何が成り立っているかを予測することがYSPである。

常識的に考えれば、当然その人は死ぬと予測されるが、ここで注意しなければならないのは動作ごとに変化する状態の情報が完全には与えられていないということである。例えば、少し待っていたときに、弾丸が何らかの理由で（例えば、弾丸が氷で出来ていて、待っている間に溶けでしまって）、銃から消えてしまって、もはや銃に弾丸が込められた状態でなくなるかもしれない。したがって、上の情報から古典論理をもちいて各状況で何が成り立っているかを決めようと思っても決めることが出来ない。古典論理を用いて結論を得たい場合には、すべての状態と動作に関して、変化するかどうかの完全な情報を与えなければならない。特に変化しないものに対しても記述しなければならないという問題をFrame問題とよぶ。

デフォルト推論やサーカムスクリプションなどの非単調推論の枠組みは、Frame問題の近似的解決の候補としてみられていた。それらの枠組みでは、推論できる変化のみが与えられた変化と考え、それ以外は変化しないとする表現によって常識を表すことで、変化しないものを記述しないですむのではないかと考えられていた。

YSPは、このFrame問題の一例として提出されたとみることができるが、それらの枠組みではwaitの間に¬ loadedになって結局 aliveになるというモデルも存在し、意図した結果が導かれないのである。この我々の常識に合わないモデルが出てくる原因と、それに対する解決策について次のような点が指摘されている[7]。

① 時間の進む方向を無視しているため、時間に沿った推論を行うモデルを選ぶメカニズムにする
② 状態変化の情報が何も与えられていないwaitでloadedが変化しているので、明示的に因果関係が与えられているところ

1	$bird:\neg injured \rightarrow fly$	
2	$bird:ist(penguin,\neg fly)$	
3	$bird\ penguin:ist(Taro,\neg injured)$	
4	$bird\ penguin:ist(Taro,\neg fly)$	
4 Neg - entering	$bird\ penguin\ Taro:\neg fly$	
1 β	$bird:injured$	$bird:fly$
3 Pos - entering	$bird\ penguin\ Taro:\neg injured$	$bird\ penguin\ Taro:\neg injured$
2 Pos - entering		$birdpenguin:\neg fly$

図4 タブロー木(例2)

で変化するモデルを選ぶメカニズムにする

③ 状態継続の公理の対偶を使ってこのモデルが生じているので、状態継続の情報を表現するときに、逆方向の推論には使えないような表現のできる推論メカニズムにする

ここでは contextual reasoning によって①の推論メカニズムが実現できることを示す。

contextual reasoning では時間を context にわりあてる。過去を外側の context に、現在を内側の context に、というような時間に沿った割り当てにより、時間をさかのぼった推論は許されない。

まずそれぞれの情報を context を用いて表すと次のようになる。

$$c_a:load \rightarrow loaded$$
$$c_a:shoot \wedge loaded \rightarrow \neg alive$$
$$c_1:alive$$
$$c_1:ist(c_2,load)$$
$$c_2:ist(c_3,wait)$$
$$c_3:ist(c_4,shoot)$$

上の2つの式は状態の変化の情報で常に成り立つ式なので、context c_a に記述している。また lifting スキーマを $ist(c_a,\varphi)\rightarrow ist(c_i,\varphi)$ $(i=2,3,4)$ とする。下の4つの式は c_1、c_2、c_3、c_4 に alive、load、wait、shoot を時間にそって割り当てている。また context 系列で状態の慣性を表現している。

すべての情報をリストアップし、質問 $ist(c_3,\neg alive)$ の否定を加えタブロー木を作成する。(図3) ふつうタブロー規則を上から順番に適用していけばよいが、ここではなるべく枝がはやく閉じて記述量が少なくなるように順番を変えてある。このタブロー木には4つの枝がでているがすべて閉じている。一番左の枝から $c_a:\neg load$ と $c_2:load$ 、 $c_a:loaded$ と $c_a:\neg loaded$ 、 $c_4:shoot$ と $c_a:\neg shoot$ 、 $c_4:alive$ と $c_a:\neg alive$ がそれぞれ矛盾している。よって $c_4:\neg alive$ がいえる。

以上、時間を context に用いる例として YSP を示してきた。この枠組みが YSP に使える理由として、①で述べたように人間と同じ思考順序で推論する体系を記述することが可能なためである。

5.2 例2 (例外処理)

例外処理の例として鳥について考える。ほとんどの鳥は死んだり、怪我をしたりしていなければ飛ぶことができる。しかしペンギンのような例外も存在する。太郎というペンギンが存在したとして次のような例を考える。

① 怪我 (injured) をしていなければ鳥は飛ぶ (fly)。

② ペンギンは鳥であるが飛ばない。

③ 太郎はペンギンで怪我をしていない。

この情報が与えられたときに、太郎は飛ぶかという質問である。

ここでは鳥、ペンギン、太郎を context に割り当てる。鳥はペンギンを、ペンギンは太郎を包含し、これらの context は入れ子構造になっている。3つの情報を context を用いて表すと

$$bird:\neg injured \rightarrow fly$$
$$bird:ist(penguin,\neg fly)$$
$$bird\ \ penguin:ist(Taro,\neg injured)$$

になる。これらの情報に太郎は飛ぶという質問の否定を加え、タブロー木を作成したものが図4になる。
左の枝では injured に関して矛盾が存在する。bird と bird penguin Taro の二つの context において Injured には lifting スキーマが適用されるとすれば、この枝は閉じている。右の枝では fly に関して bird と bird penguin Taro の context で、また bird と bird penguin の context でそれぞれ矛盾している。しかし fly に関しては bird と bird penguin の context の間では lifting スキーマが成り立つが、それ以外の context の間では成り立たないと考えるべきである。bird penguin で fly が否定されているので、bird での fly が bird penguin には継承されないためである。よって枝は閉じていなく、太郎は飛ぶということは成り立たないことがわかる。

6. おわりに

context の命題論理に対して意味論を定義し、それに基づいて文脈推論システムにおける質疑応答のための決定手続きとして prefix 付タブロー法を検討した。Massacci[8]は同様の方法を提案し、その計算量が NP-Complete であることを示している。本論文では、より簡単なタブロー規則を示し、また lifting スキーマとの関係について検討した。

lifting スキーマは context を扱って推論していくにあたり重要な役割を果たしている。ある context で成り立つものが別の context でも成り立つという lifting スキーマは prefix 付タブロー法による証明手続きの中に無理なく導入することができる。例1のように context を時間順に割り当てる場合は、常に成り立つものについては lifting スキーマが成り立つと考えて良いが、例2のような場合には注意が必要である。lifting スキーマが成り立つかどうかを明確に表現するためには、情報が継承されない場合、

$$bird:fly$$
$$bird\ \ penguin:\neg fly$$

のように否定を記述するのは当然であるが、

$$bird:fly$$
$$penguin:\neg fly$$

というように context 系列を fly に限って bird を省いて記述した方が prefix 付タブロー証明の中で使う場合を考えるとわかりやすいかもしれない(context 系列で情報の継承を表現しているので bird の context を省くことにより、fly という情報の継承はなくなる)。

今後の課題としては、一つは context を入れ子構造のものに限らなくても使えるようなタブロー規則の記述の工夫があげられる。これを実現することにより、はじめにあった速度のようなまったく別の状況の推論を扱うことができるようになる。二つ目に lifting 公理について、もっと判別しやすい記述または方法を考えることがあげられる。

参考文献

[1] W. Lukaszewicz, Non-monotonic reasoning:Formalization of commonsense reasoning,Ellis Horwood,1990.

[2] 鈴木浩之，"状況理論の基礎概念，" 人工知能学会誌, Vol.7, No.3, pp.385-391, 1992.

[3] 中島秀之，"状況を対象とした推論，" 人工知能学会誌, Vol.5 No.5 pp.588-594, 1990.

[4] J. McCarthy, " Notes on formalizing Context, " Proc. of IJCAI-93, pp.555-560, 1990.

[5] M.Fitting, "Basic modal logic," in D. M. Gabby, C. J. Hogger and J. A. Robinson, editors, Handbook of Logic in Artifical Intelligence and Logic Programming, Vol. 1 Logical foundations, pp.365-448, Oxford University Press, 1993.

[6] J. McCarthy and P. J. Hayes, "Some philosophical problem from the standpoint of artifical intelligence," Machine Intelligence , Vol.4, pp.463-502,1969.

[7] 佐藤健 ,Yale Shooting 問題とその解決へのアプローチ，” 人工知能学会誌， Vol.3, No.2，pp.132-138，1998.
[8] F. Massacci, “Contextual Reasoning is NP-complete,”AAAI-96， pp.621-626，1996.