TM-0865

任意時間推論にむけて－意味情報を用いた抽象化

有馬　淳，石川幹人

February, 1990

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

ICOT-TM

任意時間推論にむけて - 意味情報を用いた抽象化

(研究メモ)

第一研究室 有馬 淳, 石川 幹人

1 はじめに

定理の自動証明は，論理に信をおく人工知能の分野で，その初めより主要な研究対象であった．しかしながら多くの努力にもかかわらず，今なお定理証明の研究は，現実的な問題を扱うには満足のいかない点を残している．それは計算コストに関わる点である．伝統的な定理証明アルゴリズムは，論理的正当性を重んじるため NP 完全以上の複雑さを持つ．ところが，実世界での推論は有限資源という制約，すなわち例えば時間的な限界を持たざるを得ない．そのため定理証明アルゴリズムでは，必要な時間内に証明を終えられないという事態が非常に頻繁に発生する．たとえ多項式時間クラスに落したアルゴリズムを適用しても，十分とは言えない．実世界の推論に要求されている課題は，ある要求された時間内に証明を終え，なんらかの結論が得られることである．さらには，時間的限界が前もって分からない場合も現実にはあり，その場合，この課題は重ねて困難なものになる．本報告では，ある有限時間内に扱うことが困難な問題を，その扱いが可能なレベルまで“小さく”し，“大抵の場合論理的正当”である解を得る，より現実的な推論法の実現手法を探る．以下に示すものは，このような動機により行った研究の初期結果報告である．

2 意味情報を使った階層的抽象化

ここで提案する手法の基本要素は 2 つからなる．意味情報を使った抽象化，および，階層的抽象化である．前者の方法を用いるについては，いくつかの述語の真偽が形式的な制約とは別に，統計的情報などの，ある種の情報の存在によって予測可能であると想定している．このような形式的表現には現れていない情報を，意味情報と呼ぶ．この意味情報を利用して与えられた知識を抽象化することで，扱いが困難な問題をより扱い易い問題に変換するのである．従来，構文情報のみに関して抽象化を行う方法は [Plaisted 81] などで報告され，領域に依存した意味情報による階層的計画立案は [Sacerdoti 74] などが知られているが，ここで提案する手法は両者をひとつの枠組で捉えている (この他に“意識性 (awareness)”に根ざしたものとして [広沢 89] がある)．また，これまで抽象化は解そのものではなく，解を得るための戦略として使用されてきた．ここでは解自身が抽象化されていることを許し，時間的猶予に応じてより完全な解が得られるという特徴を持つ手法を考えたい．そのために意味情報を利用し，段階的に抽象化された複数の知識 (公理) を利用する．

この報告では実例を通じて簡単に概説する．まず，以下の一般的知識に関する公理の集合 A が入力されたとする．

```
(not_fly(X) :- not_ab1(X)),
(ab1(X) :- bird(X)),
(ab1(X) :- plane(X)),
(fly(X) :- not_ab2(X), bird(X)),
```

```
(ab2(X) :- penguin(X)),
(not_fly(X) :- not_ab3(X),penguin(X)).
```

この公理は先頭から，“通常のものは飛ばない”，“鳥はその例外”，“飛行機もその例外”，．．．，を表している (ここで，“not_”は論理否定を表す)．この一般ルールとその例外を書き分ける方法は，我々の知識をアプリオリに完全には記述できない (qualification ploblem) という考え方に基づき J.McCarthy が示した方法である [McCarthy 86]．この表現方法は公理の修正なしに例外を追加できる利点がある．ここで“Tweety はペンギンであり，鳥である”ことを表す，Tweety という個体に関する公理 F，

```
bird(tweety),
penguin(tweety).
```

および“Poly は鳥である”ことを表す，Poly という個体に関する公理 F'，

```
bird(poly).
```

を考える．A に F を加え，Tweety は飛ぶかどうかの質問 Q を行い，A に F' を加え，Poly は飛ぶかどうかの質問 Q' を行うとしよう．人間の常識的な推論を想像すると，ペンギンである Tweety は飛ばず，鳥である Poly は飛ぶと結論されるのが妥当である．ところがこのままでは，質問 Q, Q' に対し，Tweety は飛ばず Poly は飛ぶとは結論されない．それが演えきされるには，“Poly は飛ぶことに関して鳥の例外ではない (not_ab2(Poly))”，“Teety は飛ぶことに関してペンギンの例外ではない (not_ab3(Tweety))”を公理に加えねばならない．しかしこれら知識を前提とすることは，飛ぶ飛ばぬをあらかじめ知ることにほとんど相当してしまうから，その公理への付加には無理がある．すでに，このような我々が普段自由に扱っている“通常成り立つルール”にまつわる問題を扱った研究は，“非単調推論”の分野で精力的に行なわれているが，ここでは，その分野で従来とってきた方法とは違った手法でそれに答える．

さて，我々が日常使用しているルールは，往々にして不完全なものである．実際にはそのルールに関連する条件が多々あり，よくよく調べてみると，条件の吟味が不十分であることが判明したりするものである．それらのなかには，普段はとうてい考える必要のない条件もあれば，時間が許せば調べるに越したことはないが，それほど重要でない条件もある．そのような条件は“無視”してしまってもうまくゆく場合がある．つまり，あらかじめ予想される評価結果に置き換えてしまえば，その条件について考える必要をなくせるのである．例えば我々は，あるものを持ち上げようとつかんだとき，それが非常に熱くとも，“さて，どうしよう?”と考えたりしないものである．即座の判断で手を放すであろう．それを落して壊してしまう危険性がなくはないが，大抵の場合うまく行くことを“生得的に”あるいは“経験的に”知っているのである．先の例題では，ある段階において“飛ぶことに関して鳥の例外ではない”という条件を無視しても“通常”間違わない．また時間的に緊迫した場合を想定すれば，あえてそうすべき段階があると考えられる．

上に述べたことを可能とするために，いくつかの述語に対する評価表を与えておく．この表の内容があらかじめ知られている意味情報に相当する．さらに，時間的猶予に合わせられるように，重要さの段階ごとにレベル分けしておく．この評価表を仮に抽象化表と呼ぶ．次のように抽象度が帰納的に定義される．

1) 与えられた公理集合は抽象度 0 である．

2) 抽象度 n-1 の公理集合に n レベルの抽象化操作を行なった結果は抽象度 n の公理集合である．

ここで，n レベルの抽象化操作とは，抽象表の n レベルに記されたすべての述語を，対応する抽象述語に置き換える評価の操作である．

次の表は先の例題に対する抽象化表であり，これに基づいて A に対し抽象化を行なった結果を各抽象度に応じて示す．

抽象化表 ：

abst.lev	predicate	abst. predicate
3	ab1(X)	false
2	ab2(X)	false
1	ab3(X)	false

抽象度 1 :

```
(not_fly(X) :- not_ab1(X)),
(ab1(X) :- bird(X)),
(ab1(X) :- plane(X)),
(fly(X) :- not_ab2(X), bird(X)),
(ab2(X) :- penguin(X)),
(not_fly(X) :- penguin(X)).
```

抽象度 2 :

```
(not_fly(X) :- not_ab1(X)),
(ab1(X) :- bird(X)),
(ab1(X) :- plane(X)),
(fly(X) :- bird(X)),
not_penguin(X),
(not_fly(X) :- penguin(X)).
```

抽象度 3 :

```
not_fly(X),
not_bird(X),
not_plane(X),
(fly(X) :- bird(X)),
not_penguin(X),
(not_fly(X) :- penguin(X)).
```

このように抽象度が高いほど公理は簡単化される．例えば，抽象度1 と 2 の公理を比べると，1 では鳥の飛ぶことに関する例外存在の可能性を加味したものになっているのに対し，2 ではその可能性を無視し，“すべての鳥は飛ぶ”とルールを抽象化している．

ここで，これら (抽象度 0 から 3 まで) のそれぞれに対し，F (あるいは F') を加え，各抽象度の公理集合へ同時に質問 Q (および Q') を発するとしよう．すると一般的に言って公理が単純なもの，即ち，抽象度が高いものほど早く応答が返ることが期待できる．また逆に，抽象度が低いほど詳しい情報を扱っているため，この意味でより正確な答えになる．そこで，ここでの推論はある時間猶予の中で返ってきた解答の内で，もっとも抽象度が低い (従って，最も正確な) 解を，全体としての解として選ぶことにする．この推論の従来にみられない特徴は，任意の時間的猶予に応じた正確さで解を返すこと，非単調な (つまり時間に応じて解が変わることがある) システムを構成することなどがあげられる．(抽象度 0 の公理に基づく質問も同時に行っていることに注意せよ)．このため，問題を解くのに使用する prover (定理証明器) が何であろうと，もしその時間猶予内で解を出せるものであれば，このシステムは必ず同じ解を出す．

下は実行結果を示している． prover には M.Stickel の一階述語論理 prover， PTTP [Stickel 88] を使用している．

Abs\	fly(tweety)		fly(poly)	
3	Yes (0 ms)	No (0 ms)	Yes (0 ms)	No (0 ms)
2	Yes (0 ms)	No (20 ms)	Yes (0 ms)	?
1	?	No (20 ms)	?	?
0	?	?	?	?

ここで "?" は証明不能のため解答がなかったことを示す．まず，Tweety の質問について考える．抽象度 3 と 2 では， Yes, No がともに証明されてしまう矛盾を呈するので，判断が下せないが，抽象度 1 では， No のみが証明されるので， "Tweety は飛ばない" と結論できる．また，抽象度 0-3, Yes/No を 8 プロセスで並列計算することを想定すると， 20ms 以前は抽象度 2 で "Tweety は飛ぶ" と暫定的に結論され， 20ms 経過時に抽象度 1 で "Tweety は飛ばない" に結論が変化し，以後は安定することになる．一方， Poly の質問について考えると，抽象度 2 で "Poly は飛ぶ" が結論される．並列計算では，終始 "Poly は飛ぶ" が安定して結論されることになる．抽象化を用いない通常の推論（抽象度 0 ）では，どちらの質問に対しても解が得られないことも，この表からわかる．

3 他の実験について

前節では，意味情報を使った階層的抽象化手法を，常識推論の例題で説明した．常識推論の他にも，行動計画や法的推論を例題にして実験している．これらについても結果を概説しておこう．

行動計画の例題のひとつは， ABSTRIPS[Sacerdoti 74] で扱ったコーヒーの問題に準じて作成された．この例題では，コーヒー豆，コーヒーミル，台所，お湯，お店，お金，銀行，持つ，買う，行くなどに関する知識 (公理) が用意され，いかにしたらコーヒーが飲めるかを計画する．次の抽象化表に基づき，コーヒーを入れる行動が任意時間推論された．

abst.lev	predicate	abst. predicate
2	possess(X,S)	true
1	at(L,S)	true

抽象レベル 1 は，任意の状態 S において任意の場所 L に行けるものと抽象化している．抽象レベル 2 は，任意の状態 S において任意のもの X を手に入れられると抽象化している．これにより，抽象度が高いと詳細な行動を積極的に無視した，簡潔な行動計画が生成される．出力結果は次の通り．

Abs\	PLAN	TIME[sec]
2	pour(coffee,drip(coffee,S))	0.04
1	pour(coffee,drip(coffee,boil(water,mill(cof_beans,buy(cof_beans,have(money,s0))))))	0.34
0	pour(coffee,drip(coffee,boil(water,go_to(kitchen,mill(cof_beans,buy(cof_beans,go_to(beans_shop,have(money,go_to(bank,s0))))))))))	1.94

抽象度 2 の出力結果は，単にコーヒーをドリップして注ぐという大雑把な計画にとどまっている．抽象度 1 では，コーヒー豆がないのが分かり，お金で豆を買い，ミルで挽いて，お湯でドリップするという本格的な計画が生成される．抽象度 0 では，銀行や豆店や台所へ行く部分が詳細化されている．実行時間も，抽象度が下がるに従って増加することが示されている．

この結果から，抽象度の高いほうから順に推論すると，次第に，より詳細化された計画が得られることがわかる．またこれは，時間に制限のある場合でも，任意の時点で，それまでの時間に見合った計画のアウトラインが用意されていることを意味する．このほか，溺れる人を助けるロボットという状況設定の行動計画問題でも，同様の任意時間性が確認できた．

法的推論についても実験を行った．法的推論のなかでもとくに時間的要素の強い陪審審議の過程をとりあげた．陪審審議は 12 人の陪審員により，全員一致の評決が得られるまで無制限に続けられる．戯曲「12 人の怒れる男」([ローズ 56]) に描かれている陪審審議の内容から例題を取った．この例題は，証拠に合理的疑問が次々と見つかり，陪審の考えが有罪傾向から無罪に変化していく法的推論過程のモデル化である．各証拠の合理性を抽象化レベルに設定し，任意時間推論でモデル化した．その結果，抽象度が下がるにしたがって，証拠の合理性が調べられるように推論が進み，だんだんと無罪評決に至る様子が良くシミュレーションできた．

4. 今後の課題

意味情報を使った階層的抽象化手法を用いた任意時間推論が，いくつかの種類の問題に適用でき，効果が挙がることが確認できた．今後は，より一層大きな問題に応用していくとともに，並列計算のより効果的な利用方法を探っていきたい．前者では，部分問題によって解答の出る抽象度が異なる場合に，どう対処していくかが課題である．後者では，早めに計算の終了する抽象度の高い計算結果を手がかりに，引続き詳細な推論を空き資源を利用して行わせることができないかが課題になる．

[参考文献]

[Plaisted 81] Plaisted,D.A.: Theorem Proving with Abstraction, Artificial Intelligence 16 p47-108 (1981).

[Sacerdoti 74] Sacerdoti,E.D.: Planning in a hierarchy of abstraction spaces, Artificial Intelligence 5 p115-135 (1974).

[広沢 89] 広沢 誠: ABR - 意識情報を用いた推論方式の提案，第 39 回平成元年後期情報処理全国大会.

[McCarthy 86] McCarthy,J.: Application of circumscription to formalizing common-sense knowledge, *Artificial Intelligence 28* 89 - 116 (1986).

[Stickel 88] Stickel,M.E.: A Prolog Technology Theorem Prover: Implementation by Extended Prolog Compiler, Journal of Automated Reasoning 4 p353-380 (1988).

[ローズ 56] 十二人の怒れる男 (額田やえ子訳)，劇書房 (1988).