

構造写像エンジンの高速化

山路 孝浩 森下 太朗 椿 和弘

シャープ(株)情報技術開発センター

1 はじめに

類推は、既知の経験や知識を基にして未知の問題を解決する際に使われる基本的な推論方式である。構造写像エンジン (Structure Mapping Engine: 以下SME) は、高次の関係構造を持つものをより重視する、数少ない類推計算プログラムの一つである。筆者らは、与えられた関係記述から対象や制約の対応関係をシステムティックに計算するSMEの機能は、事例ベース推論、特に事例想起(検索)に利用可能であると考え、簡単な例題を題材として研究を進めている[2]。しかしながら、SMEを用いて事例検索を行う場合、構造の大きな事例に対しては膨大な処理時間を要するという問題が生じている。これを解決するために、高速化のアルゴリズムを考案し、実験を行った。本論文では、そのアルゴリズムと実験結果について報告する。

2 構造写像エンジンの概要と処理の流れ

SMEは、2つの系(ベースとターゲット)の間の対応を調べるプログラムである。木構造の関係表現により表されるベース、ターゲットが与えられると、各系の評価値付きの対応関係を計算し、ターゲットに不足している関係(推論候補)をベースから推論する。図1はSMEの処理を表したものであり、ピーカーとガラス瓶の水の流れ(ベース)から棒で接続されたアイスクューブとコーヒー(ターゲット)に熱の流れがあるということを推論している。

基本的な処理の流れは次の4プロセスからなり、[1]に従っている。

- ① 局所マッチ構築: 潜在的マッチ可能ペアをみつけ、マッチ仮説を作る。
- ② Gmap構築: マッチ仮説を一貫性が保たれるように結合し、最大のものを作る。なお、一貫性とはGmap内のベースとターゲットの要素が1対1に対応することを意味する。
- ③ 推論候補構築: 各Gmapの推論候補を導き出す。
- ④ Gmapの評価点付け: 各Gmapに評価点を付ける。

3 負荷の原因

2における①~④プロセスの内、②の負荷が非常に重い(全体の実行時間の約9割)。②のプロセスは下記のサブプロセスから成り、負荷の大部分は、サブプロセス③の処理による。

① 2の①で、出来たマッチ仮説に対して一貫性を調べる。

② ベースに着目し、同一構造下にあるもので、一貫性を保つGmapは結合する。

③ ベースの構造に関係なく、一貫性を保つGmapは結合する。

③の部分、例えば、図1の例における (LIQUID(water) * LIQUID(coffee), water * coffee) と (FLAT_TOP(water) * FLAT_TOP(coffee), water * coffee) (3の①の結果生成されたGmap)を組み合わず処理に相当する。

③の処理を言い換えると与えられたN個の集合から、ある条件を満足する全ての異なる集合の組み合わせを求める処理ということになり、典型的な組み合わせ問題となっている。[1]に忠実に実現すると、各Gmapを結合することに他要素との一貫性を調べるということになり、一貫性が保たれなくなるまで繰り返すことになる。単純にこれに従うとチェック回数は2^Nオーダーとなり、また一貫性チェック自体の負荷も大きく、莫大な処理時間を要する原因となる。

4 高速化アルゴリズム

一貫性について、一般的に、次のことが成り立つ。A, B, C, DはGmapとする。

$$\begin{aligned} & (A, B, C, D) : \text{一貫} \Leftrightarrow \\ & (A, B), (A, C), (A, D), (B, C), (B, D), (C, D) : \text{一貫} \quad (*) \end{aligned}$$

この多項間の一貫性は2項間の一貫性に置き換えられることに着目して、下記のアルゴリズムを採用した。

(i) 3の②で生成されたGmapに対してGmap間同士の一貫性を調べ、一貫しているペアを作る。

A fast algorithm for the structure mapping engine.
Takahiro YAWAJI, Taro MORISHITA, Kazuhiro TSUBAKI,
SHARP Corp. Information Systems Research and Development Center

例

A, B, C, D, E, F: Gmap. {X, Y}: XとYは一貫しているとする。

{A, B}, {A, C}, {A, E}, {A, F} (a)

{B, C}, {B, E}, {B, F} (b)

{C, F} (c)

{D, E}, {D, F} (d)

{E, F} (e)

(i) (e) を利用し、(i) で生成されたペアを(a)から2項間で組み合わせていく。

(ii) 包含関係にあるGmapを消去する。例えば、{A, C, F} は {A, B, C, F} に含まれているので消去する。

このアルゴリズムは、以前のアルゴリズムに比べて、多項間の一貫性チェックを行う必要がないことが特徴である。

現在、このアルゴリズムにより評価を進めている。その中から実験結果の一部を表2に示す。例3はベースにもターゲットにも同じ名前の述語が多く存在する。ベースにおける本構造の多くは深さが2で、一つしか木の枝をもたない。Gmap間の結合(4の(i))を調べると37個中33個までが28個のGmapと一貫性があり、残りのGmap4個は30個のGmapと一貫性がある。これが原因となり、Gmap結合部(3の③)の負荷が重く、生成されるGmap数が多くなると考えられる。例4は本構造の深さはほとんどが3で、中には深さ5の本構造が存在し、木の枝はほとんど二つ以上ある。構造的には例3よりも大きい。一方、Gmap間の結合(4の(i))を調べると、各Gmapは平均15個のGmapと一貫性がある。例3と比較すると結合しうるGmap数が少ない。その分、Gmap結合部(3の③)の負荷が軽く、生成されるGmap数や処理時間も少なくなっていると考えられる。実験結果より、Gmap結合部(3の③)の負荷の重さ、生成されるGmap数や処理時間の多きなどは、Gmap数(3の②)の結果生成されたGmapや構造の大きさよりも、結合しうるGmap数の比率に依存すると考えられる。なお、例3および例4の測定できずはメモリー不足のため計測できなかったことによる。

4 おわりに

高速化を試みたが、まだ、構造が大きいものに対しては上述の(i)および(ii)による負荷が大きい。そこで、対策として並列化と最適解のみ求めるアルゴリズムを現在検討中である。なお、本研究は第5世代コンピュータプロジェクトの一環として行われているものであり、本研究の機会を与えて頂いたICOT第7研究室の新田室長に深く感謝いたします。

[参考文献]

[1] Falkenhainer, B. "The structure mapping engine", In proceeding of International Association for

Artificial Intelligence, 1986.

[2] 柿和弘、他、構造写像エンジンを利用した設計支援の考察、情報処理学会第43回全国大会、1991。

[3] 森下太朗、他、事例ベース推論における構造写像エンジンの利用、情報処理学会第42回全国大会、1991。

[4] 森下太朗、他、構造写像を利用した事例ベース推論システム、人工知能学会第17回知識ベースシステム研究会資料、1991。

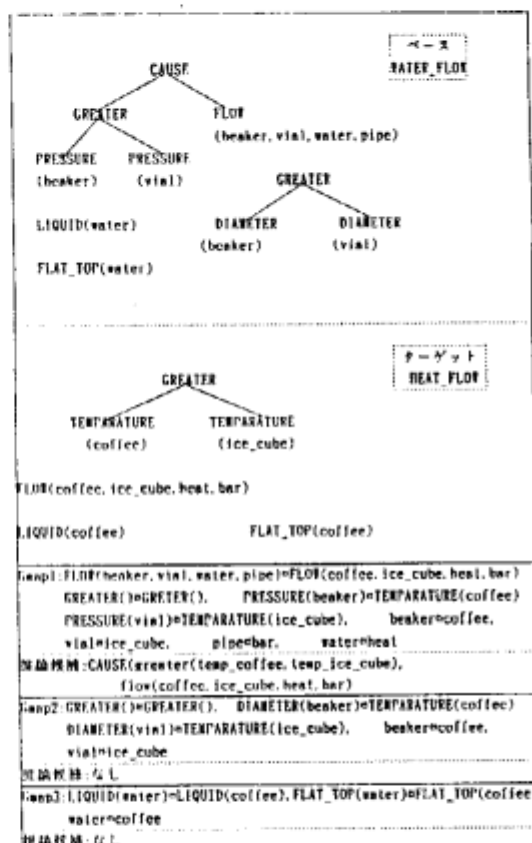


図1 SWE処理概要(入出力)

	Gmap数 (3の②の結果)	生成された Gmapの数	改良前(msec)	改良後(msec)
例1	18	180	2069013	6649
例2	19	240	3820849	9362
例3 h: 2 l: 8	37	2880	測定できず	519576
例4 h: 2 l: 7	33	120	測定できず	48967

表2 アルゴリズムの測定結果

(注) 時間は、Gmap結合部のみ(3の③)にかかった時間である。表に記載されているh, lとはベース、ターゲットのことであり、数字は、本構造の数を表している。