1P－2

実時間知識処理をめざした制約推論における木探索制約充足手法のレスポンスタイム推定法

遠城　秀和
NTTデータ通信（株）開発本部

1　はじめに

近年の知識処理技術の進歩により、プロセス制御やオンライントランザクション処理のような実時間処理分野においても知識を使ったより複雑な処理の実現が求められている。これらの実時間処理においては、レスポンスタイムなどの時間条件を満たす保証がシステム実現上必要である [1]。しかし、従来の知識処理では、答えが有限時間内に求められる保証がなく、与えられた時間内に答えを求めるというレスポンスタイムの保証をすることができなかった。このため、実時間処理分野への知識処理技術の応用が困難であった。

本報告では、知識処理の一推論方法である制約推論法の木探索制約充足手法において、質問を与えてから答えが求まるまでのレスポンスタイムを知識の量と計算機の処理性能で表現し、レスポンスタイスを推定する方法について述べる。

2　実時間環境における知識処理のレスポンスタイム

従来のエキスパートシステムなどの知識処理では、利用者などから質問を受け付け、質問に対する推論結果を答として返すという処理が主な処理の流れになる。一方、実時間処理ではオンライン処理のトランザクションやプロセス制御のセンサデータなどが外部イベントとして入力され、そのイベントの処理結果が出力として返される。したがって知識処理を実時間処理に導入するには、外部イベントを知識処理への質問に変換し (入力変換処理)、推論結果をそのイベントに対する出力に変換する (出力変換処理) ことで可能となる。(図 1)

実時間処理では、外部イベントの入力時からイベントに対する返事を返すまでを通常のレスポンスタイムとしている。それと同様に、実時間環境における知識処理では外部イベントの入力時から、入力変換処理、推論処理、出力変換処理を順番に実行し、返事を返すまでをレスポンスタイムとすることになる。

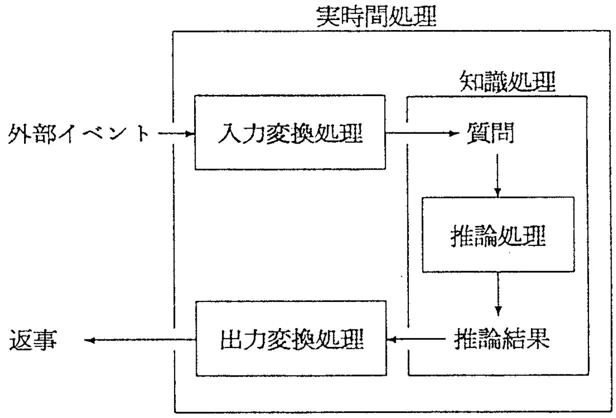

図 1: 実時間環境における知識処理

したがって知識処理のレスポンスタイムを保証するためには、処理全体に与えられた時間を分割し、決められた各時間内に入力変換処理、推論処理、出力変換処理が終了する保証が必要である。この内、入力変換処理と出力変換処理は従来の実時間処理のモジュールとして実現できるため、従来と同じ方法で与えられた時間内の終了を示すことが可能である。このため、推論処理が与えられた時間内の終了を示すことが、知識処理のレスポンスタイムを保証する上で重要である。

したがって、実時間環境における知識処理のレスポンスタイムは、入力変換処理の処理時間、推論処理の処理時間と出力変換処理の処理時間の和として推定できる。

3　推論処理における計算量

推論処理が与えられた時間内の終了を証明するには、推論処理時間に上限が存在しなければならない。一般に、処理時間の上限は計算量理論により解析されている。これまでに知識処理の推論方法としては、ルール、論理、セマンティックネット、フレーム、制約などが提案されている。しかし、計算量の解析が十分に行なわれていない。その中では制約推論の計算量が比較的解析されており、NP 完全という結論ではあるが処理時間の上限が示

Responce Time Prediction for Tree Method of Constraint-Based Inference toward Real-Time Artificial Intelligence
Hidekazu ENJO
NTT DATA COMMUNICATIONS SYSTEMS CO.

されている [2] [3]。しかし、計算量のオーダが示されているだけでは、レスポンスタイムを推定するためには十分な精度と言えない。レスポンスタイムの推定に用いるには、係数の値も含めた完全な計算式の形での計算量推定法が必要である。

制約推論の解法は、列挙法、木探索法、弛緩法、併合法の4種類に分類できる [2]。列挙法における計算量推定方法は、既に筆者による報告 [4] があり、併合法における計算量推定方法は、窪田らによる報告 [5] がある。ここでは、木探索法を用いた推論処理における計算量の推定方法を示す。

4 木探索法制約推論の推論処理時間

木探索法は縦型探索を基にした制約充足手法であり、幾つかに分割された部分問題を順番に、そして既に解いた部分解を用いながら解いていく方法である。ここでは、逐次的に部分問題を解きながら全ての解を求める問題の推論処理時間の算出方法を示す。

逐次的に部分問題を解く場合、全体の推論時間 (Tp) は各部分問題を推論する時間 (T_j) の和になる。部分問題の数を N とおくと、以下の計算式になる。

$$Tp = \sum_{j=1}^{N} T_j \tag{1}$$

各部分問題を推論する時間は、既に解いた部分解一つごとに全ての候補解を生成し制約条件で検証する時間になる。部分解一つごとに生成される候補解は、部分解に含まれない部分問題中の新規変数が取ることのできる値の全ての組み合わせと部分解を合わせたものになる。したがって、部分解一つごとに生成される候補解の数は、新規変数が取る得る値の個数 (V_{jk}) の積になる。与えられる部分解の個数を L_j 、新規変数の個数を n_j とすると各部分問題で生成される候補解の数 (M_j) は、以下の式で計算できる。

$$M_j = (\prod_{k=1}^{n_j} V_{jk}) L_j \tag{2}$$

一つの候補解を検証する時間 (t_j) は、候補解を生成する時間 (T_g) とその候補解を各制約条件ごとに検証する時間の和になる。制約条件を基本的な検証処理の形に分解しておくことで、一つの候補解を一つの制約条件で検証する時間は一定と考えることができる。(この時間を単位検証時間 (T_c) とする。) このため、一つの候補解を各制約条件ごとに検証する時間の和は、制約条件の数 (m_j) に比例することになり、制約条件の数と基本検証時間の積になる。したがって、一つの候補解を検証する時間は以下の式で計算できる。

$$t_j = m_j T_c + T_g \tag{3}$$

部分解の個数は、その部分問題の制約条件の強さに依存し一定の値にはならない。しかし、部分解の個数は部分解を解く際に検証した候補解の個数を上回らない。このため部分解の個数の上限を候補解の個数とすることができる。部分解個数の不等漸化式から以下の式になる。

$$L_j \le \prod_{l=1}^{j-1} \prod_{k=1}^{n_l} V_{lk} \tag{4}$$

一つの候補解の検証時間は各候補解の値に依存しない。また、部分解一つごとに生成される候補解の個数も部分解の値に依存しない。したがって、各部分問題を推論する時間は、各部分問題で生成される候補解の数と一つの候補解を検証する時間の積になる。したがって全体の推論時間の上限は以下の式になる。

$$T_p \le \sum_{j=1}^{N} (m_j T_c + T_g)(\prod_{l=1}^{j} \prod_{k=1}^{n_l} V_{lk}) \tag{5}$$

木探索法制約推論を用いたレスポンスタイム (T_a) は、入力変換の処理時間を T_i 、出力変換の処理時間を T_o とすると以下の式で計算できる。

$$T_a \le T_i + (\sum_{j=1}^{N} (m_j T_c + T_g)(\prod_{l=1}^{j} \prod_{k=1}^{n_l} V_{lk})) + T_o \tag{6}$$

5 まとめ

本報告では、知識処理の一推論方法である制約推論法の木探索制約充足手法において、候補解の生成数と制約条件数を知識の量とし、一つの候補解を生成する処理時間と一つの制約条件を検証する処理時間を計算機の処理性能として、質問を与えてから答えが求まるまでのレスポンスタイムを推定する方法について述べた。

参考文献

[1] JIS X 0010-1987 情報処理用語 (操作技法及び機能)

[2] 西原: "整合ラベリング問題と応用", 情報処理, Vol.31, No.4, pp500-507 (1990)

[3] Dechter, R. and Pearl, J.: "Network-Based Heuristics for Constraint-Satisfaction", Artificial Intelligence, Vol.34, No.1, pp1-38 (1990)

[4] 遠城: "実時間知識処理をめざした制約推論のレスポンスタイム推定法", 第44回情処全大, 2Q-4 (1992)

[5] 窪田、内野、李、山下、西原: "併合法による制約充足の並列化効果について", 第46回情処全大, 7A-1 (1993)