

2Q-4

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

遠城 秀和

NTTデータ通信(株) 開発本部

1. はじめに

プロセス制御やオンライントランザクション処理の分野のような実時間環境においても知識処理技術の進歩により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、レスポンスタイムが保証されていることがシステム実現上必要である。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というレスポンスタイムの保証がされていなかった。このため、実時間処理分野へ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において、質問を与えてから答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、レスポンスタイム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、利用者などからの質問を受け、それに対する推論結果を答として返すという動作が主な処理の流れとなっている。また実時間処理では、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レスポンスされる。したがって実時間環境に知識処理を導入するには、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)、推論結果をそれらイベントに対する出力として変換すること(出力変換処理)で可能となる(図1)。

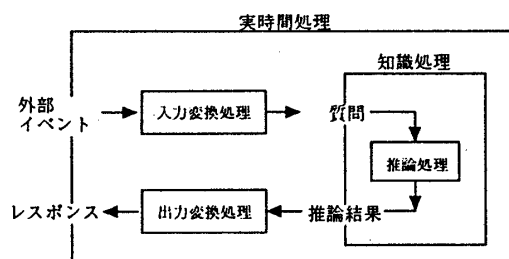


図1 実時間環境における知識処理

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レスポンス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のレスポンスタイムは、外部イベントの入力時からイベントに対する出力をレスポンスするまでとなり、入力変換処理の時間、推論処理の時間、出力変換処理の時間の和が知識処理のレスポンスタイムとして定義できる。

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、入力変換処理、推論処理、出力変換処理の各処理が有限時間内に終了することが証明されることが必要となる。入力変換処理、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、従来と同じ方法で有限時間に終了できることを示すことが可能である。このため、推論処理が有限時間内に終了できることを示すことが、知識処理のレスポンスタイムを示す上で重要である。

3. 推論処理における計算量

推論処理が有限時間内に終了することを証明するには、推論処理時間に上限が存在しなければならない。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手法としては、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が、これまでに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されている[1][2]。しかし、これまでに示されている処理時間の上限はオーダが求められているのみであり、処理時間の推定としては用いることができない。そのためレスポンスタイムを推定するに十分な精度の計算量推定方法が必要である。

制約推論の解法としては、列挙法、木探索法、弛緩法、併合法の4種類に分類される[1]。ここでは、解法として最も単純な列挙法を用いた場合の推論処理時間の計算量を推定することとした。

4. 列挙法制約推論における推論処理時間

列挙法は、全ての候補解を生成し、生成した候補解が制約条件を満たすかどうかテストして解を求め

する方法である。一つの候補解は、与えられた制約条件に用いられている各変数に、その変数が取り得る値の中から一つの値に定めて生成される。したがって候補解の全生成数は、各変数の取り得る値の全ての組み合わせ数、すなわち各変数の取り得る値の数の全ての積になる。

テストは生成した各候補解が制約条件をみたすかどうかを調べる処理である。具体的には、生成した候補解が与えられた制約条件を一つ一つ満たすかどうかを調べることになる。したがって、生成した一つの候補解をテストする処理では、一つの制約条件を調べる処理を与えられた制約条件の数だけ行うことになる。このため、生成した一つの候補解をテストする処理時間は一つの制約条件を調べる処理時間と制約条件数の積に一つの候補解を生成する処理時間を加えた時間になる。

したがって列挙法による制約推論の処理時間 (T_p) は、生成される候補解数 (N_s)、制約条件数 (N_c)、一つの候補解を生成する処理時間 (T_g)、一つの制約条件を調べる処理時間 (T_c) を使って表現すると次式になる。

$$T_p = (T_g + (T_c \times N_c)) \times N_s$$

また候補解を一つ生成する時間は、各変数の値を定めるだけであるので非常に短いと仮定することができる。このため、制約条件を満たすかどうかをテストする時間に比べ、候補解を一つ生成する時間は無視することが可能である。したがって、列挙法による制約推論の処理時間の近似式として、次式が得られる。

$$T_p = T_c \times N_c \times N_s$$

列挙法制約推論を用いた知識処理のレスポンスタイム (T_a) は、入力変換の処理時間 (T_i)、出力変換の処理時間 (T_o) と組み合わせて表現すると次式になる。

$$T_a = T_i + (T_g + (T_c \times N_c)) \times N_s + T_o$$

また近似式は以下のようになる。

$$T_a = T_i + T_c \times N_c \times N_s + T_o$$

5. 制約推論処理のレスポンスタイム推定

ここでは、レスポンスタイムを推定する例題としてテトリスというコンピュータゲームを用いた。テトリスの様に計算機側がユーザの入力に関係なく動的に状況を変化させるゲームをする場合、ユーザは必要な操作を与えられた適当な時間内に行わなければならない。このため計算機によってテトリスを行うには、レスポンスタイムを保証することが必要となり、実時間環境の知識処理の応用と考えることが

できる。

入力変換の処理は、積み上げられているブロックの状況と落ちてくるブロックの種類と回転位置からどこに落せばよいかという質問への変換になる。出力変換の処理は、現在位置と落す位置の関係に従って落下してくるブロックを回転させたり左右に移動させる操作列への変換になる。

テトリスを行う知識処理としては、(1)「ブロックの下に空きを作らないで、一番高さが低い所に置く」、(2)「必ず空きができてしまう時は真下に置く」の2つの規則を用いた。テトリスで用いるブロックは7種類であるが、回転した場合を区別すると28個の制約条件によって記述できている。また制約条件は、行と列という2つの変数を用いて表現している。さらに横10ブロック、縦30ブロックの枠のテトリスを想定しているので、生成される候補解は300個となる。

入力変換の処理時間は小さいと仮定でき、出力変換処理もそれほど時間がかかるわけではないが入力変換よりは処理時間がかかると仮定した。そこで、入力変換の処理時間と出力変換の処理時間と和 ($T_i + T_o$) を、1秒とした。また一つの候補解を生成する処理時間と一つの制約条件を調べる処理時間それぞれ1m秒とした。

列挙法制約推論を用いた知識処理のレスポンスタイム推定式を用いると、 $T_a = 1000 + (1 + 1 \times 28) \times 300$ となり、推定値は9.7秒である。したがって、10秒程度の余裕があれば、この方法で計算機がテトリスをすることは可能と判断できる。

6. 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において、候補解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調べ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が求まるまでのレスポンスタイムを表現する方法を示した。さらに、テトリスを例として用い、レスポンスタイムを推定する方法を述べた。

参考文献

- [1] 西原清一：整合ラベリング問題と応用, 情報処理 Vol.31, No.4, pp500-507, (Apr. 1990)
- [2] Dechter, R. and Pearl, J. : Network-Based Heuristics for Constraint-Satisfaction, *Artificial Intelligence*, Vol.34, No.1, pp1-38, (1988)