

開放型プロダクションシステムにおけるデータ依存関係の管理

2K-9

西部喜康 横尾真 石田亭

NTT 情報通信処理研究所

1 まえがき

近年、エキスパートシステムの応用領域として監視、制御などの動的に変化する実世界を対象とすることが考えられている。動的に変化する実世界を対象とした場合、ワーキングメモリの内容は推論中においても実世界の変化を反映し、変更する必要がある。しかしながら、エキスパートシステムの推論機構として一般に用いられているプロダクションシステムでは、推論中に他のプロセスがワーキングメモリ内のデータを変更しないことを前提としている。

本稿では推論中に他のプロセスがワーキングメモリの内容を変更することを前提として推論を行なうプロダクションシステム(開放型プロダクションシステムと呼ぶ)を提案する。開放型プロダクションシステムにより推論途中に実世界の変化を反映することが可能となるが、この場合推論で生成したデータと、実世界の変化の反映にともなって作られたデータとの間に矛盾が生じることが問題となる。このためワーキングメモリ内に矛盾が生じないように管理する必要が生じてくる。

本稿では、ワーキングメモリ内のデータの依存関係およびその成立条件を管理することで無矛盾性を維持する方法を報告する。

2 従来の手法と問題点

開放型プロダクションシステムでは、実世界の変化に実時間対応するために推論途中でもワーキングメモリ内のデータの変更を行なうが、この時ワーキングメモリ内に矛盾が生じないように管理することが必要になる。我々はまず次に示す方法で無矛盾性の管理を行なうことを考えた。

データの論理的依存関係(データが存在するための根拠: データの生成の原因になったデータの情報)を保持し、あるデータが削除もしくは変更された場合、そのデータに依存していたデータを全て削除し、再び推論を再開する。

この方法では、矛盾を持つ可能性があるデータ、すなわち変更前のデータに基づいて生成させたデータはすべて削除するためワーキングメモリ内に矛盾が生じるのを防ぐことができる。

また、この方法はエキスパートシステム構築支援ツール ART[Inference Corporation, 1987] のデータ依存関係管理機能を開放型プロダクションシステムに適用することに相当する。しかしながらこの方法によれば、データの依存関係だけを用いるため、次のような問題が起ってくる。

- 1) データの変更が依存関係に影響しない場合でも、その変更されたデータに依存していたデータは削除される。

- 2) この場合、削除されたデータはルールの発火によって再生成される。

例えば $(a=10), (b=20)$ というデータがあり、 $a+b$ が 50 以下なら $(c=100)$ というデータが作られるという場合があったとする。 $(a=10), (b=20)$ から $(c=100)$ が作られた後に $(a=10)$ が $(a=20)$ に変化したとする。データ依存関係管理機能では、いったん $(c=100)$ というデータは削除され、ルールの発火により再生成される。

当然、上記で指摘した削除、および再生成の処理は無駄な処理であり、高速化の観点から見てマイナスである。

このデータ依存関係管理機能を持ったプロダクションシステム上で、代表的な例題の 1 つである Monkey-and-Bananas(ブランディングを行ない、そのプランを基に行動するように記述されている)を動作させたところ、ルールの発火によってデータが変化するたびに、変化したデータをもとに作っていたプランを削除し、再生成すると言う無駄な処理が認められた。この無駄なルールの発火数は全発火数 22 の内、5 ルールであった。

3 条件つきデータ依存関係の導入

本稿では削除、再生成の無駄な処理を省くために、データの依存関係だけでなくその成立条件も考慮したデータの変更処理を方法を提案する。本方式ではデータの依存関係だけでなく成立条件も同時に保持しており、ワーキングメモリの更新処理において依存関係の成立条件を評価し、データの依存関係を再構築することにより、無駄な処理を減少させることを可能とすることである。この成立条件つきデータ依存関係管理機能は、一般的にワーキングメモリに対して行なうデータの生成、追加、削除、変更処理に付加機能を付け加えることで実現できる。

上記の成立条件に基づいたデータの処理を行なうためには以下の処理が必要とされる。

- 1) データの生成、追加では依存関係だけでなく、その成立条件も蓄える。
- 2) データの削除は、データ依存関係管理機能と同等の処理を行なう。
- 3) データの変更時は各データの依存関係の成立条件を参照することで依存関係を調べ、成立条件が

Data Dependency Maintenance For Open Production System.

Yoshiyasu NISHIBE Makoto YOKOO Toru ISHIDA

NTT Communications and Information Processing Laboratories

崩れる場合はデータの削除を行ない、依存関係に影響がない場合は依存関係の変更を行なう。

上記の更新処理をワーキングメモリに対して行なえば、データの変更が成立条件に影響しない場合、無駄なデータの削除、再生成の処理を防ぐことが可能である。

4 条件つきデータ依存関係の管理

本章では、前章で述べた処理のうち特に変更の処理について、例を用いて説明する(図1参照)。データを変更する際には2通りの場合が存在する。データを変更することにより依存関係が崩れる場合と、保たれる場合である。図中、円で示してあるのはデータ、各データをつなぐ線はデータの依存関係を示す依存リンクを表し、依存リンク上に四角形で示してあるのが依存関係の成立条件である。

ワーキングメモリがこのような状態にある時、データCをC'に変更したとする。

1) データの変更で成立条件が崩れる場合

データCをC'に変更し、その後データCに依存していたデータを全て削除する。(ここではEがその対象となる)。その結果、ワーキングメモリの状態は図1(a)の様な状態に変化する。

2) データの変更で成立条件が崩れない場合

まず、データC'を単独で生成する。続いて依存関係に関するデータのうちCの情報をC'に変更する。これによりEはC'およびDにC₂という条件で依存することになる。その後Cについて削除の処理を施すことによりC-A、C-B、E-C間の依存関係は削除する。この状態を示したのが図1(b)である。

上記のような機能を有するプロダクションシステムを試作し、その上でMonkey-and-Bananasを動作させた。成立条件を考慮しないデータ依存関係管理機能では、動作の終了までのルールの発火数は先に述べた通り22であった。一方、本稿で提案した成立条件つきデータ依存関係管理機能を有するプロダクションシステム上で動作させた場合は理想値17で終了した。これにより、成立条件に基づく処理により無駄な処理を減少可能なことが確認された。また、実行時間に関しては、成立条件つきデータ依存関係管理機能を使用した方が約20%短い時間で処理を終了した。

5 むすび

成立条件つき依存関係に基づいたワーキングメモリ更新処理を行なうことにより、データの無矛盾性を維持し、しかも無駄な処理を減少させることができとなる。本方式では、データの変更が依存関係に影響を及ぼさないものであれば依存リンクを張り直すことにより、変更前のデータに依存していたデータに対して削除の処理を行わない様にしている。

本機能は、開放型プロダクションシステムの様なワーキングメモリ内のデータが頻繁に変化する場合や、ワーキングメモリ内のデータが複雑な依存関係を

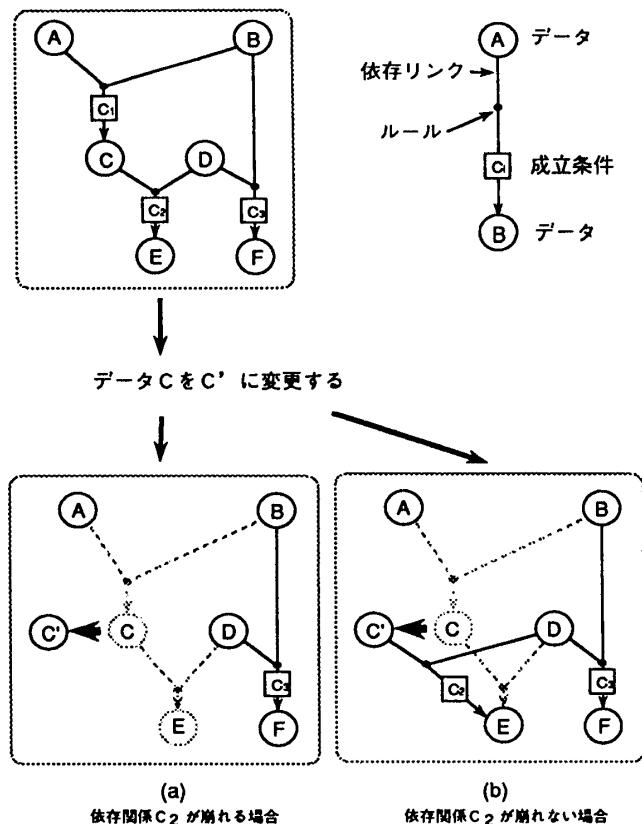


図1: データの変更例

持つ場合に処理量を削減させるという意味で有効である。

なお、本稿ではデータ駆動型プロダクションシステムに対しての手法を提案してきたが、より単純なデータ構造を持つform-filling型プロダクションシステム(同一クラスのデータのインスタンスが1つに限定され、あらかじめ与えられている)については、Schaeferらが同等の機能を実現している [Schaefer et al., 1986]。

[参考文献]

[Inference Corporation, 1987] Inference Corporation, ART Reference Manual, 1987

[Schaefer et al., 1986] Philip R. Schaefer, Ishi H. Bozma and Randall D. Beer, Knowledge-Based Validity Maintenance For Production Systems, AAAI-86, pp. 918-922, 1986