

7G-5

知識メディアステーション

(3) 対話的問題解決環境

熊谷 秀光 和氣 朝臣 瀧口 伸雄

三菱電機(株)中央研究所

1. はじめに

知識メディアステーションにおける「セッション」は、表、文書、グラフ、図形、イメージ等の知識と推論機構を結び付けるためのユーザインタフェースであり、以下の特長を持つ。

- 柔軟な問題解決を行えるように制約指向論理型言語^{[1],[2]}を知識プログラミング言語に採用した。
- 制約指向論理型言語の問合せ機構として、Incremental QueryやInteractive Query Revisionを拡張した対話的問合せ機構を導入した。
- 表、文書、グラフ、図形、イメージと推論機構との入出力が推論内容と独立に行える入出力インタフェースを導入した。

本論文では、知識メディアステーションのセッションにおける対話的問題解決環境について述べる。

2. 論理型言語と対話的問合せモデル

一般的に、論理型言語を用いた問題解決は、問題領域を論理式で記述し、解くべき条件を問合せとして与え、その解を推論により求めることである。代表的な論理型言語である Prologの場合、問合せ、すなわち解の満足する条件は論理積の関係で結ばれたゴール列で与える。

このようなPrologでの問題解決の方法は、一つのゴール列に問題を解くための条件を完全に記述しなければならない点で人間の試行錯誤的問題解決には向かない。例えば、図1で、二つのゴール節(1)、(2)の共通の変数名「_太郎の親」に何も関係がない。(1)、(2)の積をゴールとして与えたいときは、(3)のような問合せを行わなければならない。

- ?- 親(太郎, _太郎の親). ... (1)
 ?- 兄弟(_太郎の親, 太郎の親の兄弟). ... (2)
 ?- 親(太郎, _太郎の親),
 兄弟(_太郎の親, 太郎の親の兄弟). ... (3)

図1 Prologの実行環境

この問題を解決するために、van Emdenは Incremental Queryと呼ばれる新しい問合せ環境を提案した^{[3],[4]}。Incremental Queryでは、問合せは段階的に入力可能でその解を見ながら人間は次に必要となる問合せを入力していくことになる。

図2の(2)は実行上「???- A1,A2,B1,B2,B3」と解釈される。人間は(1)と(2)の問合せによる解を見ながら、段階的に足りない条件を補ったり、現在の結果を否定するような問合せを対話的に行うことにより、他の解やより詳細な解を求めることができる。

Incremental Queryではこのような複数の問合せ間の変数による関係付け(問合せレベルでの大域変数の導入)、

???- A1,A2. ... (1)

S1 S1:問合せ(1)の解

???- B1,B2,B3. ... (2)

S2 S2:S1と問合せ(2)の解の合成

図2 Incremental Query

および問合せをまたぐバックトラックのほかに、入力した問合せの部分的な削除を行うことができる。例えば、図2の問合せB2を削除して、問合せを「???- A1,A2,B1,B3」に変形することができる。M.OhkiらによるInteractive Query Revisionは、Incremental Queryの考えをさらに発展させ、問合せの追加、削除、置き換え等の問合せの編集機能を強化したものである^[5]。

3. 制約指向論理型言語

一般に論理型言語における問合せモデルでは、問合せの順序を考慮しなければならない。例えば、図3のような算術演算では、Cを求めるために(1)の計算を先立って行う必要がある。Prologのような論理型言語ではこのような論理式、算術式の順序が存在し、問合せを記述するときにユーザは問題解決手順を考慮しなければならない。

A = B+1. ... (1)

C = A*A. ... (2)

図3 算術演算における順序

このような順序を考慮しながら問合せを発することは、Incremental Query等を用いた柔軟な問題解決を妨げる要因となり得る。そこで我々は、推論エンジンとして制約指向論理型言語を採用した。制約指向論理型言語は、計算可能な条件が満たされるまで実行を保留したり、連立方程式等を自動的に解いたりするメカニズムを論理型言語に付加しており、この結果問合せの順序を意識する必要がなくなる。この意味で、制約指向論理型言語は Incremental Query等の対話的問合せモデルとProlog以上に親和性がある。

また対話的問合せ機構としてIncremental Query等に以下のような拡張を加えたものを導入した。

- 問合せの編集環境としてスクリーンエディタを用い、問合せを一度に複数編集可能にすることによって、効率的な編集をできるようにした。
- 異なったゴール節間でのバックトラックと同じゴール節内でのバックトラックを統一的に扱うために、修正時にどの問合せから実行するかを保存するテーブルを導入した。
- 現在の問題解決状況を表示するモニタを備えることによって、制約指向論理型言語による問題解決の可能性を高めた。

このような制約指向論理型言語を用いた環境により、計算機と人間の能力を互いに補いながら問題の解決策を考えていく対話的問題解決が可能になった。

4. 問合せ環境

3. で制約指向論理型言語による問題解決について述べたが、問題解決の途中においては問合せの状態を把握できることが望ましい。これは、制約指向論理型言語の実行環境において問合せと問合せ結果が同次元のものではなく、また制約指向特有の保留条件（例えば式の数が少ないために解くことができない連立線形方程式）等が存在するためである。この点についてセッションでは、以下に述べるような環境を提供することにより問題解決過程の可視化を行っている（図4）。

1) 問合せウィンドウ

問合せウィンドウは問合せを入力するウィンドウである。これは3. で述べたようにスクリーンエディタになっている。

2) 保留条件ウィンドウ

保留条件ウィンドウは、条件が揃わないために実行を保留している問合せや内部のゴール（解の不定な連立方程式を含む）を表示するウィンドウである。

3) 変数ウィンドウ

変数ウィンドウは問合せに用いた変数名を表示するウィンドウである。

4) 問合せ参照ウィンドウ

問合せ参照ウィンドウは変数名を指定することにより、その変数の値及びその変数を含む全ての問合せを表示する。

5) 変数領域ウィンドウ

変数領域ウィンドウは、変数の取り得る値の範囲を表示するウィンドウである。

6) 様々な形態の知識との入出力インタフェース

人間との親和性の高い問題解決を行うために、表、文書、グラフ、図形、イメージ等と入出力を行うための機構を備えている。

図形、イメージを取り扱うための環境を備えている。推論機構とこれらの知識はエントリ変数と呼ぶ特別な変数名を持つ変数でつながっている^[6]。例えば、ある表のB列3行目の欄は、その表がウィンドウ1であるとき、「S1B3」という変数名で参照できる。また文書の場合は、文中の任意の領域を参照する変数を定義でき、グラフや図形の場合は任意の作図パラメータに対応する変数が定義でき、さらにイメージの場合は任意の領域を参照する変数を定義できる。このような変数を問合せ中で用いることにより、表や文書等の中から変数の具体値を取ったり、表や文書等へ具体化された変数の値を書き込んだり、その値によって図を描いたりすることが自動的に行われる。これはセッションが問合せで用いられている変数名を管理し、表や文書等を参照している変数とそうでないものを区別することによって実現されている。

このような変数の値の入出力機構は、表や文書等にデータを書き込むことにより推論を起動できること、また多彩な形式での解の表示が容易であること、さらにまた表に表示したものを棒グラフに書き換えるといったインタフェースの交換が容易にできるといった利点を持つ。

またセッションではマウスを用いて、アイコンやメニューを選択するといった統一した操作で知識の生成を行なえるようにし、簡単に知識を用いることができる環境を提供している。また、知識はマルチウィンドウによって表示され、複数の知識を同時に見ながら問題解決を進めていくことができる。

4. で述べた問合せの編集環境を用いることによって、効率よく足りない条件や変数を補ったり、システムの状態を監視することができ、デバッグ作業を含めて能率的なシステム開発を進めていくことができる。

6. おわりに

本論文では、知識メディアステーションの「セッション」における対話的問題解決を行う環境について述べた。論理型言語における対話的問題解決は、人間の長である試行錯誤的思考法にマッチしたものである。また、様々な知識表現が可能なこと人間と計算機との協調的問題解決に果たす役割は非常に大きい。

今後、高速化かつ柔軟な推論機構、洗練された知識表現等の方式を検討し、より人間にとって親和性の高いインタフェースを研究して行く予定である。

参考文献

- [1] Catherine Lassez: Constraint Logic Programming, Byte, August(1987), pp.171-pp.176
- [2] Alain Colmerauer: Opening the PrologIII Universe, Byte, August(1987)pp.177-pp.182
- [3] 淵一博監修, 古川康一・溝口文雄共編: インタフェースの科学, 共立出版(1987)
- [4] M.H.van Emden: Logic as an Interaction Language, Proc. of 5th Conf. Canadian Soc. for Computational Studies in Intelligence(1984)
- [5] M.Ohki, A.Takeuchi, K.Furukawa: A Framework For Interactive Problem Solving based on Interactive Query Revision, Proc. of Logic Programming Conference '86(Tokyo)(1986)
- [6] 中島他: 知識メディアステーション (2) 知識プロセッサ, 情報処理学会第37回全国大会(1988)

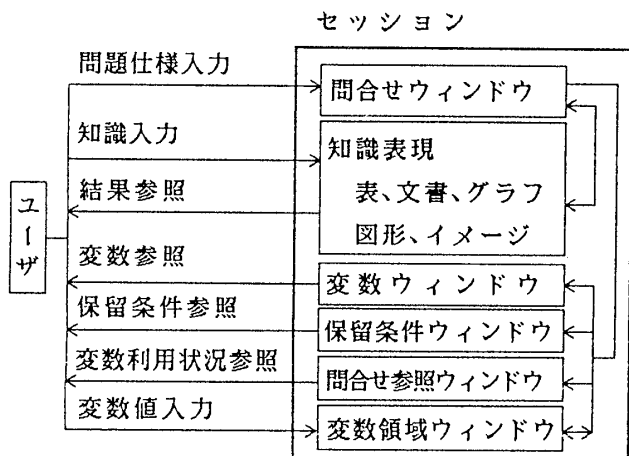


図4 セッションのインタフェース構成

5. セッションによる問題解決

知識メディアステーションの制約指向論理型言語は入出力のための特別な機構を持たないという長を持つ。これはプログラムとその入出力インタフェースとの独立性を高める。このことは解の表示方法を考えずに問題解決の方法を記述できると共に、後で適切なインタフェースを適宜用意することによって、分かりやすい解の表示および条件の入力を行えることに結び付く。また、簡単に複数のインタフェースを同一問合せ中で用いることが可能になる。

セッションは4. で述べたように、表や文書、グラフ、