PROLOGのHyperCardインタフェース

5J-7

渋谷 正弘 (北海道工業大学 工学部)
田中 譲 (北海道大学 工学部)

1.はじめに

論理型プログラミング言語Prologの操作性を向上させるためにHyperCardとの融合を試みたので報告する。Prologは、言語仕様が単純であるためにプログラムを記述し易く、新たな機能の追加も容易にできる言語である。しかし、従来のProlog処理系のインタフェースは、あまり使い易くないためにプログラミングしずらくなっている。そこで、操作性の良いApple社製のHyperCard[1]をPrologのインタフェースとして利用することでPrologの親和性の向上を試みた。

HyperCard上でPrologを実行することにより、HyperCardのプログラミング言語であるHyperTalkではプログラミングすることが困難であった文字の演算や、再帰プログラミング、解が一意に決定できない問題などを解くことが可能となる。これにより、HyperCard上でエキスパートシステムなどを作成することも容易となる。また、ファクトやルールなどは、HyperCard上で1つずつカード単位で取り扱うことができる。カード上に書き表されたファクトやルールは、取り扱いも容易で直感的に理解し易い。

2.HyperCardのインタフェースをもつProlog

Prologを用いてプログラミングを行うには、Prologのデータベース部分にファクトやルールを登録し、次に質問文を評価させて解答を得るという手順をとる必要がある。効率よくプログラミングを行うには、登録しておいたファクトやルールを簡単に参照でき、編集できる必要がある。従来のProlog処理系では、ファクトやルールの参照や編集をするには組み込み述語を利用する必要があった。ユーザがエディタを利用して新たにプログラムを作成したり既にあるプログラムを編集する際、Prologの実行モードからエディタのモードにモードを変更し、エディタのコマンドを利用してテキストを作成した後、Prologの実行モードにこの作成したテキストを格納するという手順を取る。また、エデットされるプログラムはロール紙に描かれたテキストの様に長くなるため、編集する箇所がまとまっていない場合には不便に感じる。これらの事柄から、以下の点を改善したシステムを作成する。

(1)Prologプログラムの実行モードと編集モードの区別をなくす。

(2)エディタでの登録、参照、編集が簡単に行えるようにする。

(3)ユーザの実行過程を記録したログが簡単に作成でき、それの再利用が容易になるようにする。

上記3点を改善するために、HyperCard中にPrologを埋め込みHyperCardを介してPrologを利用できるスタックを作成した。HyperCardは、ビル・アトキンソンらによって作成された文字、絵、音などのいろいろな情報をカードの中に埋め込んで利用するツールである。カード上でユーザが行う動作は、ボタンを押す、フィールドやメッセージボックスに文字をタイプする、メニューからツールあるいはオブジェクトを選ぶなどの簡単な動作より成り立っているので操作性に優れている。ユーザによって蓄積された情報は、カード上に保存されているので直感的に理解し易いなどの特徴を持つ。

HyperCard上でPrologを利用するために、Prologのプログラムや質問文などの文字情報をHyperCardのフィールド内で利用できるようにした。また、Prologプログラムの実行過程を記録したログもカード上に記録できるようにした。HyperCard上でPrologプログラムの作成から実行、修正までを行えるシステムを作成することにより、モードをなくすことができた。3章にシステムの全体像を示す。

Interfacing Prolog with HyperCard

Masahiro SHIBUYA[1], Yuzuru TANAKA[2]
1:Hokkaido Institute of Technolgy 2:Hokkaido University

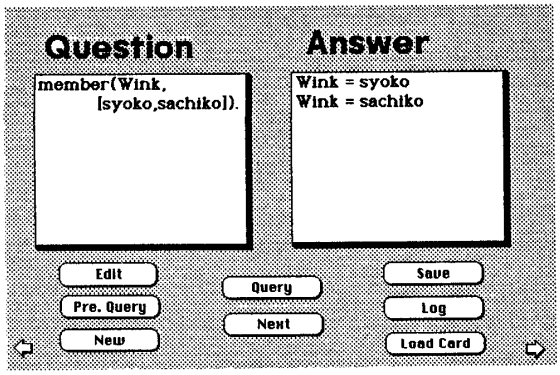

図1 述語"member"の実行例

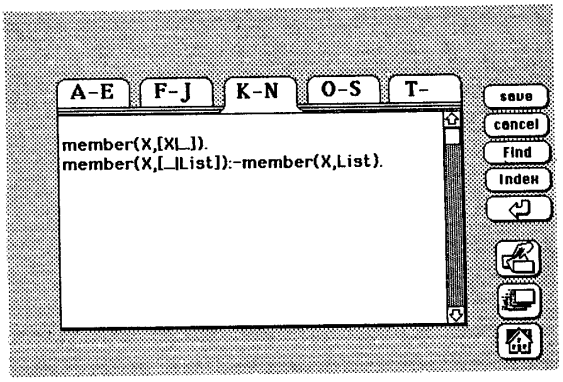

図2 述語"member"のエディット例

3.システム構成

HyperCard上でPrologが稼働するシステムのメイン画面を図1に示す。ユーザがこのシステムを利用するには、左側のフィールド欄に質問文を入力し、"Query"ボタンを押すと右側のフィールド欄に質問文に対する解答が表示される。解答が他にもある場合は、"Next"ボタンを押せば次の解答が得られる。この質問文とその答えが表示されているカードは"Save"ボタンにより保存される。また、"Log"ボタンを押しておくことによりPrologシステムとのやり取りの様子がカードに記録される。また、プログラムを編集するには"Edit"ボタンを押しエディットカードを呼び出す。

エディットカードを図2に示す。エディットカードでは、ユーザが定義した述語を1つずつカード上に登録する。新規に登録された述語カードはアルファベット順にカードに登録されるので、後で参照する時にはカード上部にあるアルファベットをマウスでクリックし、目的の述語を見付ける。また、既に登録されている述語を一覧するには"Index"ボタンを押せばよい。このエディタは、登録された述語を1つずつ表示しているが従来のエディタの様にロール紙状に表示し、編集することも可能である。

このシステムは、HyperCardとPrologのデータをやり取りを行うXCMDをHyperCardに追加することにより実現されている。そのため、図1、2に示した画面はユーザが自由に再定義や修正が可能である。

4.まとめ

HyperCardで稼働する言語HyperTalkは、カード上で定義されたオブジェクトを制御することに重点を置いた言語である。そのために、HyperTalkを他のプログラミング言語のように利用する時、プログラミングができなかったり、速度が遅いなどの問題が生じる。このような場合には、XCMDやXFCNと呼ばれる機械語プログラムによってコマンドや関数を追加しなければならない。HyperCardを用いて意志決定支援システムを作成する場合、HyperTalkをXCMDやXFCNによって拡張し、記述するよりもPrologで記述したほうが容易に作成できる。

ユーザは、Prologを用いて一連のまとまった動作を扱うメッセージハンドラの内容を定義するシステムを作成することもできる。このシステムは、ユーザが作成してほしいメッセージハンドラの特徴をキーワードとして与え、それをPrologを用いて構文解釈して、HyperTalkプログラムに変換してスクリプトに書き込むシステムである。このように、HyperCard上でPrologプログラムが実行できるならば、HyperCardの応用範囲が広がる。

また、Prologプログラムの実行過程のトレースにもHyperCardが利用できる。Prologプログラムを実行する際、ユーザが予めセットしておいた幾つかのブレークポイントの情報をカード上に書き出しておくと、実行後にこのカードを取り出し参照しブレークポイントの状況を把握できる。このように、PrologにHyperCardのインタフェースを与えることによりPrologは使い易すくなり、HyperCardの応用範囲は広がる。今後は、本システムを用いたエキスパート・システムや推論システムの研究を行う予定である。

参考文献

[1] "HyperCard User's Guide",Apple Computer ,Inc. (1987)