

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-148501

(P2000-148501A)

(43) 公開日 平成12年5月30日 (2000.5.30)

(51) Int.Cl.⁷

G 0 6 F 9/44

識別記号

5 2 0

F I

G 0 6 F 9/44

テマコード* (参考)

5 2 0 J

審査請求 有 請求項の数 8 F D (全 8 頁)

(21) 出願番号 特願平10-328834

(22) 出願日 平成10年11月4日 (1998.11.4)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 稲本 直太

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100093595

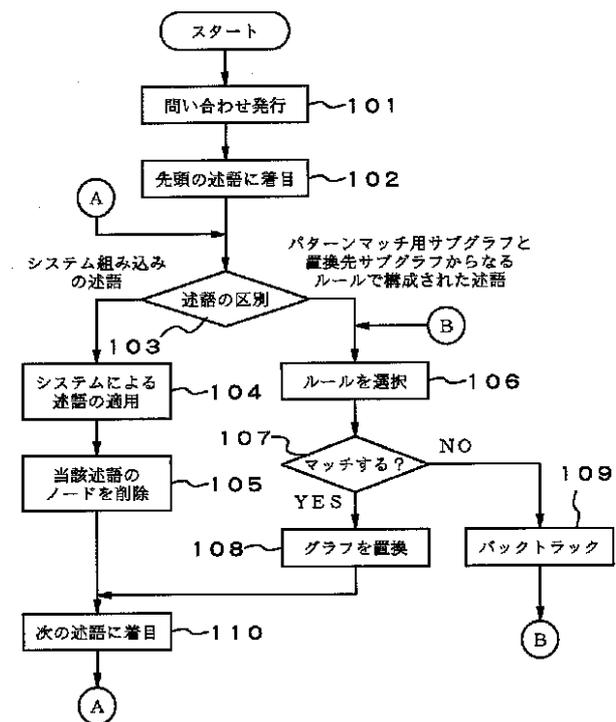
弁理士 松本 正夫

(54) 【発明の名称】 グラフ構造データを用いた論理型言語処理系の実現方法及びコンピュータ制御プログラムを格納した記憶媒体

(57) 【要約】

【課題】 グラフ構造データを用いることにより、従来の論理型言語処理系で複雑になっていた変数のバインディングを直感的に分かりやすいものにし、論理型言語処理系の実現を容易にするグラフ構造データを用いた論理型言語処理系の実現方法を提供する。

【解決手段】 論理型言語の処理系を実現する論理型言語処理系の実現方法において、論理型プログラムの述語に相当するルール及び問い合わせをグラフ構造データで表現する工程と、問い合わせを表すグラフ構造データの述語がシステム組み込みの述語である場合に、該述語をシステムが適用する工程と、問い合わせを表すグラフ構造データの述語がシステム組み込みの述語でない場合に、ルールを表すグラフ構造データのパターンマッチングと置換とを行うことにより、該述語の適用を行う工程とを含む。



【特許請求の範囲】

【請求項1】 論理型言語の処理系を実現する論理型言語処理系の実現方法において、論理型プログラムの述語に相当するルール及び問い合わせをグラフ構造データで表現する工程と、前記問い合わせを表す前記グラフ構造データの述語がシステム組み込みの述語である場合に、該述語をシステムが適用する工程と、問い合わせを表す前記グラフ構造データの述語がシステム組み込みの述語でない場合に、前記ルールを表す前記グラフ構造データのパターンマッチングと置換とを行うことにより、該述語の適用を行う工程とを含むことを特徴とするグラフ構造データを用いた論理型言語処理系の実現方法。

【請求項2】 前記システムによる述語の適用工程と前記グラフ構造データのパターンマッチング及び置換による述語の適用工程とを、前記問い合わせを表す前記グラフ構造データの先頭の述語から順次繰り返すことを特徴とする請求項1に記載のグラフ構造データを用いた論理型言語処理系の実現方法。

【請求項3】 前記グラフ構造データのパターンマッチング及び置換による述語の適用工程において、複数の前記ルールを表す前記グラフ構造データを用いたパターンマッチングと置換とを並列に実行することを特徴とする請求項1または請求項2に記載のグラフ構造データを用いた論理型言語処理系の実現方法。

【請求項4】 前記ルール及び問い合わせをグラフ構造データで表現する工程において、述語を表すルートノードと該述語の引数を表すリーフノードとからなる高さ1のツリーであって、一つのルールが、問い合わせとのパターンマッチングを行うために用意されたサブグラフと、パターンマッチングに成功した問い合わせのグラフ構造データを置換するサブグラフとの対で構成されたグラフ構造データにより、前記ルール及び前記問い合わせを表現することを特徴とする請求項1に記載のグラフ構造データを用いた論理型言語処理系の実現方法。

【請求項5】 コンピュータシステムのデータ処理装置を制御して、論理型言語の処理系を実現するコンピュータ制御プログラムを格納した記憶媒体において、論理型プログラムの述語に相当するルール及び問い合わせをグラフ構造データで表現する工程と、前記問い合わせを表す前記グラフ構造データの述語がシステム組み込みの述語である場合に、該述語をシステムが適用する工程と、問い合わせを表す前記グラフ構造データの述語がシステム組み込みの述語でない場合に、前記ルールを表す前記グラフ構造データのパターンマッチングと置換とを行うことにより、該述語の適用を行う工程とを含むことを特徴とするコンピュータ制御プログラムを格納した記憶媒

体。

【請求項6】 前記システムによる述語の適用工程と前記グラフ構造データのパターンマッチング及び置換による述語の適用工程とを、前記問い合わせを表す前記グラフ構造データの先頭の述語から順次繰り返すことを特徴とする請求項5に記載のコンピュータ制御プログラムを格納した記憶媒体。

【請求項7】 前記グラフ構造データのパターンマッチング及び置換による述語の適用工程において、複数の前記ルールを表す前記グラフ構造データを用いたパターンマッチングと置換とを並列に実行することを特徴とする請求項5または請求項6に記載のコンピュータ制御プログラムを格納した記憶媒体。

【請求項8】 前記ルール及び問い合わせをグラフ構造データで表現する工程において、述語を表すルートノードと該述語の引数を表すリーフノードとからなる高さ1のツリーであって、一つのルールが、問い合わせとのパターンマッチングを行うために用意されたサブグラフと、パターンマッチングに成功した問い合わせのグラフ構造データを置換するサブグラフとの対で構成されたグラフ構造データにより、前記ルール及び前記問い合わせを表現することを特徴とする請求項5に記載のコンピュータ制御プログラムを格納した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、Prologに代表される論理型言語処理系を実現する論理型言語処理系の実現方法に関し、特にグラフ構造データを用いた論理型言語処理系を実現する論理型言語処理系の実現方法及びコンピュータシステムのデータ処理装置を制御して当該グラフ構造データを用いた論理型言語処理系の実現方法を実行するコンピュータ制御プログラムを格納した記憶媒体に関する。

【0002】

【従来の技術】Prologに代表される論理型言語の処理系における述語の適用を行う場合、従来の論理型言語処理系の実現方法では、変数の値を変数名で表されるシンボルとバインディングさせていた。しかし、プログラム中には同じ変数名が何回も現れるため、推論が当該変数部分に至る度に変数名を表すシンボルと実際の値とのバインディングが変更される。このため、変数のバインディング処理が複雑になり、論理型言語処理系の実現に手間を要していた。

【0003】また、上記のように、従来の論理型言語処理系の実現方法は、変数のバインディングの処理が複雑であったため、実現された論理型言語処理系が非常に複雑になっていた。したがって、当該処理系による論理型言語プログラムの実行において、オーバーヘッドが生じやすく、性能が出難くなっていた。

【0004】

【発明が解決しようとする課題】上述したように、従来の論理型言語処理系の実現方法は、変数のバインディングが非常に複雑であるため、論理型言語処理系の実現に手間を要するという欠点があった。

【0005】また、変数のバインディングが複雑であることにより、実現された論理型言語処理系が複雑になるため、当該処理系による論理型言語プログラムの実行において性能が出難いという欠点があった。

【0006】本発明の目的は、グラフ構造データを用いることにより、従来の論理型言語処理系で複雑になっていた変数のバインディングを直感的に分かりやすいものにし、論理型言語処理系の実現を容易にするグラフ構造データを用いた論理型言語処理系の実現方法及び当該論理型言語処理系の実現方法を実行するコンピュータ制御プログラムを格納した記憶媒体を提供することにある。

【0007】また、本発明の他の目的は、上記の目的に加え、論理型言語処理系を簡単にするることにより、十分な性能の出る論理型言語処理系を実現することができるグラフ構造データを用いた論理型言語処理系の実現方法及び当該論理型言語処理系の実現方法を実行するコンピュータ制御プログラムを格納した記憶媒体を提供することにある。

【0008】

【課題を解決するための手段】上記の目的を達成する本発明は、論理型言語の処理系を実現する論理型言語処理系の実現方法において、論理型プログラムの述語に相当するルール及び問い合わせをグラフ構造データで表現する工程と、前記問い合わせを表す前記グラフ構造データの述語がシステム組み込みの述語である場合に、該述語をシステムが適用する工程と、問い合わせを表す前記グラフ構造データの述語がシステム組み込みの述語でない場合に、前記ルールを表す前記グラフ構造データのパターンマッチングと置換とを行うことにより、該述語の適用を行う工程とを含むことを特徴とする。

【0009】請求項2の本発明のグラフ構造データを用いた論理型言語処理系の実現方法は、前記システムによる述語の適用工程と前記グラフ構造データのパターンマッチング及び置換による述語の適用工程とを、前記問い合わせを表す前記グラフ構造データの先頭の述語から順次繰り返すことを特徴とする。

【0010】請求項3の本発明のグラフ構造データを用いた論理型言語処理系の実現方法は、前記グラフ構造データのパターンマッチング及び置換による述語の適用工程において、複数の前記ルールを表す前記グラフ構造データを用いたパターンマッチングと置換とを並列に実行することを特徴とする。

【0011】請求項4の本発明のグラフ構造データを用いた論理型言語処理系の実現方法は、前記ルール及び問い合わせをグラフ構造データで表現する工程において、

述語を表すルートノードと該述語の引数を表すリーフノードとからなる高さ1のツリーであって、一つのルールが、問い合わせとのパターンマッチングを行うために用意されたサブグラフと、パターンマッチングに成功した問い合わせのグラフ構造データを置換するサブグラフとの対で構成されたグラフ構造データにより、前記ルール及び前記問い合わせを表現することを特徴とする。

【0012】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照して詳細に説明する。

【0013】まず、本発明の一実施形態によるグラフ構造データを用いた論理型言語処理系の実現方法の前提となるグラフ構造データの構成について説明する。

【0014】本実施形態に用いるグラフ構造データは、論理型言語によるプログラムの述語に相当するルール及び問い合わせをツリー型のグラフ構造データで表し、サブグラフのパターンマッチングとその置換によって当該論理型言語の述語の適用を進める。

【0015】さらに具体的に説明すると、一つの述語は、高さ「1」のツリーである。ツリーのルートノードは述語を表し、ツリーのリーフノードは述語の引き数を表す。引き数は、変数の場合と定数の場合とがある。

【0016】述語に相当するノード（ルートノード）は、属性として述語の識別子を持つ。また、述語に相当するノード（ルートノード）と引き数に相当するノード（リーフノード）との間のアークは、属性として引き数の識別子を持つ。

【0017】述語はルールの集合からなり、一つのルールは、パターンマッチングを行うために用意されたサブグラフ（以下、パターンマッチ用サブグラフと称す）と、パターンマッチ用サブグラフとのパターンマッチングに成功した他のグラフ構造データ（問い合わせを表すグラフ構造データ）を置換するサブグラフ（以下、置換先サブグラフと称す）との対で構成される。パターンマッチ用サブグラフは、述語を一つだけ持つ。置換先サブグラフは、述語を一行に並べて構成される。

【0018】パターンマッチ用サブグラフと置換先サブグラフで共通の引き数は、同じリーフノードで表される。また、置換先サブグラフにおける共通の引き数は、同じリーフノードで表される。

【0019】問い合わせを表すグラフ構造データは、述語を一行に並べた構造であり、問い合わせを表すグラフ構造データにおける共通の引き数は、同じリーフノードで表される。

【0020】以上のように構成されたグラフ構造データを用いて、図5に示すプログラム及び問い合わせを表現した例を図2ないし図4に示す。図5は、論理型言語の一例であるPrologにより、プログラムと問い合わせとを記述した例を示す。また、図2は、図5のプログラムの1行目のルールを本実施形態で用いるグラフ構造

データの形式で表現した例を示し、図3は、図5のプログラムの2行目のルールを本実施形態で用いるグラフ構造データの形式で表現した例を示し、図4は、図5の問い合わせを本実施形態で用いるグラフ構造データの形式で表現した例を示す。

【0021】次に、上記のように構成されたグラフ構造データを用いて論理型言語処理系を実現する手順について説明する。図1は、本実施形態によるグラフ構造データを用いた論理型言語処理系の実現方法を示すフローチャートである。

【0022】図1を参照して、上述したグラフ構造データの形式で表現された問い合わせが発行されると(ステップ101)、まず、当該問い合わせにおける一列に並んだ述語の先頭の述語に着目する(ステップ102)。そして、先頭の述語がシステムに組み込みの述語である場合は、システムが当該述語の適用を行う(ステップ103、104)。システムに組み込みの述語が適用されると、当該述語を表すノードは不要になるので削除する(ステップ105)。

【0023】問い合わせの先頭の述語がパターンマッチ用サブグラフと置換先サブグラフからなるルールで構成されている場合は、まず、一つのルールを選択し(ステップ103、106)、当該述語の高さ「1」のツリーが当該ルールのパターンマッチ用サブグラフとマッチするかどうか調べる(ステップ107)。マッチする場合は、当該高さ「1」のツリーを置換先サブグラフに置き換える(ステップ108)。

【0024】バックトラックが必要な場合は、グラフ構造データで表現された問い合わせをコピーして述語を適用し、述語の一つのルールの適用後に問い合わせの結果が失敗した場合に、次のルールを元のグラフ構造データに適用することによって推論を進めていく(ステップ109)。

【0025】上記ステップ103ないしステップ109の動作を、各述語に対して順次行う(ステップ110)。以上の動作において、推論の途中に現れる変数は、サブグラフの置換によって現れたノードとして表現される。したがって、シンボル名と値をバインドさせる処理は不要となる。

【0026】次に、図5に示したプログラム及び問い合わせのグラフ構造データ(図2ないし図4参照)を用いて本実施形態による具体的な述語の適用の例を示す。図示の例では、リスト[1]とリスト[2,3]をappendした結果であるリスト[1,2,3]が出力される。

【0027】図5に示すプログラムは、本実施形態におけるグラフ構造データの表現により、図2及び図3に示すように表される。プログラムは、2つのルール、ルール(1)とルール(2)とからなり、図2及び図3を参照すると、各ルールは、パターンマッチさせるグラフ構

造データ(パターンマッチ用サブグラフ)とルールを適用した後に置換するグラフ構造データ(置換先サブグラフ)との対になっている。

【0028】図2及び図3で「append」という名前のついたノードから出ている、「x」、「y」、「z」という名前のついたアークの先のノードは、それぞれ、述語appendの第1引数、第2引数、第3引数を表す。また、「cons」という名前のついたノードから出ている、「h」、「t」、「r」という名前のついたアーク先のノードは、それぞれ、述語consの第1引数、第2引数、第3引数を表す。

【0029】本実施形態におけるプログラムの内部表現では、節中の一つの変数は一つのノードで表され、変数の参照は述語を表すノードから当該変数を表すノードにアークが引かれていることで表現される。また、引き数を表すノードの は未定義変数、 は既定義変数を表す。

【0030】以上のようなプログラムに対して、図5に示す問い合わせが発行されたものとする。当該問い合わせは、図4に示すグラフ構造データで表される。

【0031】まず、図4の破線で囲われた部分に着目し(図1、ステップ101、102参照)、この部分にルール(1)を適用しようとする(図1、ステップ103、106、107参照)。しかし、図2及び図4を参照すると、パターンマッチ用サブグラフ中の「x」という名前のついたアークの先のノードの値が、問い合わせのグラフ構造データでは[1]であり、ルール(1)のパターンマッチ用サブグラフでは[]であって、マッチしないので、ルール(1)は適用できない(図1、ステップ109参照)。

【0032】次に、ルール(2)を適用しようとする(図1、ステップ103、106、107参照)。図3及び図4を参照すると、ルール(2)は適用することができる。そこで、当該ルール(2)を適用すると、図4のグラフ構造データは、パターンマッチ用サブグラフを置換先サブグラフに置き換えて、図6のように変わる。このとき、変数部分のパターンマッチの結果、未定義変数だったappendの第1引数、第2引数を表すノードが[1],[2,3]の値を持った既定義変数に変わる(図1、ステップ108参照)。

【0033】次に、図6のグラフ構造データにおいて、破線で囲われた部分の述語の適用を行い(図1、ステップ103~105参照)、さらに、リスト[1]を1と[]に分解することにより、グラフ構造データが図7に示す形に変わる(図1、ステップ110参照)。

【0034】同様の操作を繰り返し、図7の破線で囲われた部分にルール(1)を適用して、グラフ構造データが図8に示す形に変わる。さらに、図8のグラフ構造データにおいて、破線で囲われた部分の述語の適用を行うことにより、グラフ構造データが図9に示す形に変わ

る。図9に示すように、この時点で、最初の問い合わせ時において未定義だった変数zに[1]と[2,3]をappendした結果[1,2,3]が入る。

【0035】以上説明したように、本実施形態によるグラフ構造データを用いた述語の適用の部分とシステムに組み込みの述語とを実装することにより、Prolog処理系と同等の処理系を実現することができる。

【0036】図10は、本発明の他の実施形態を示す。本実施形態は、パーソナルコンピュータやワークステーション、その他のコンピュータシステムのデータ処理装置10を上記グラフ構造データを用いた論理型言語処理系の実現方法を実行する制御プログラムにて制御することにより、Prolog等の所望の論理型言語の処理系を実現する。制御プログラムは、図示のプログラム記憶媒体20に格納して提供され、データ処理装置10にロードされて上記グラフ構造データを用いた論理型言語処理系の実現方法を実行する。

【0037】プログラム記憶媒体20としては、磁気ディスクや光ディスク、半導体メモリ、その他の一般的な記憶媒体を用いることができる。

【0038】以上好ましい実施の形態をあげて本発明を説明したが、本発明は必ずしも上記実施の形態に限定されるものではない。例えば、上記実施形態では、ルール(1)とルール(2)とを順番に試しているが、複数のルールを同時に適用して並列に推論を行うことも可能である。従来の論理型言語処理系の実現方法は変数のバインディングが並列処理に不向きであったが、本発明は、複数のルールを適用した後のグラフ構造データは独立なので、並列に推論を行うことに適している。

【0039】

【発明の効果】以上説明したように、本発明のグラフ構造データを用いた論理型言語処理系の実現方法及び当該論理型言語処理系の実現方法を実行するコンピュータ制御プログラムを格納した記憶媒体によれば、変数のバインディングを直感的に分かりやすい形で表現するため、論理型言語処理系の実現を容易にすることができるという効果がある。

【0040】その理由は、本発明では、グラフ構造データのサブグラフパターンマッチングとサブグラフの置換*

*及びシステム述語の適用によって推論を行い、推論の途中で現れる変数は、サブグラフの置換により新しくできたノードという形で表されると共に、一度既定義変数になった変数を表すノードにおいて、変数の値が変わることがないためである。

【0041】また、実現された論理型言語処理系が簡単になるため、十分な性能の出る論理型言語処理系を実現することができるという効果がある。

【図面の簡単な説明】

10 【図1】 本発明の一実施形態によるグラフ構造データを用いた論理型言語処理系の実現方法を示すフローチャートである。

【図2】 本実施形態に用いられるグラフ構造データの例を示す概念図であり、図5のプログラムの一つのルールの内部表現を示す概念図。

【図3】 本実施形態に用いられるグラフ構造データの例を示す概念図であり、図5のプログラムの他の一つのルールの内部表現を示す概念図。

20 【図4】 本実施形態に用いられるグラフ構造データの例を示す概念図であり、図5の問い合わせの内部表現を示す概念図。

【図5】 本実施形態による処理の対象となる論理型言語によるプログラム及び問い合わせの例を示す図である。

【図6】 本実施形態により図4のグラフ構造データを処理した状態を示す概念図である。

【図7】 本実施形態により図6のグラフ構造データを処理した状態を示す概念図である。

30 【図8】 本実施形態により図7のグラフ構造データを処理した状態を示す概念図である。

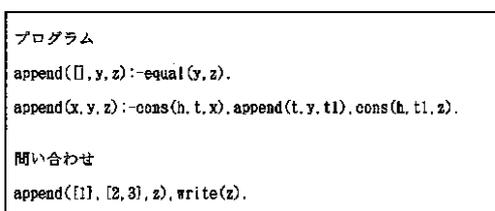
【図9】 本実施形態により図8のグラフ構造データを処理した状態を示す概念図である。

【図10】 本発明の他の実施形態による、グラフ構造データを用いた論理型言語処理系の実現方法を実行するコンピュータシステムの構成を示すブロック図である。

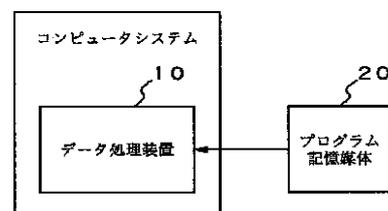
【符号の説明】

- 10 データ処理装置
- 20 プログラム記憶媒体

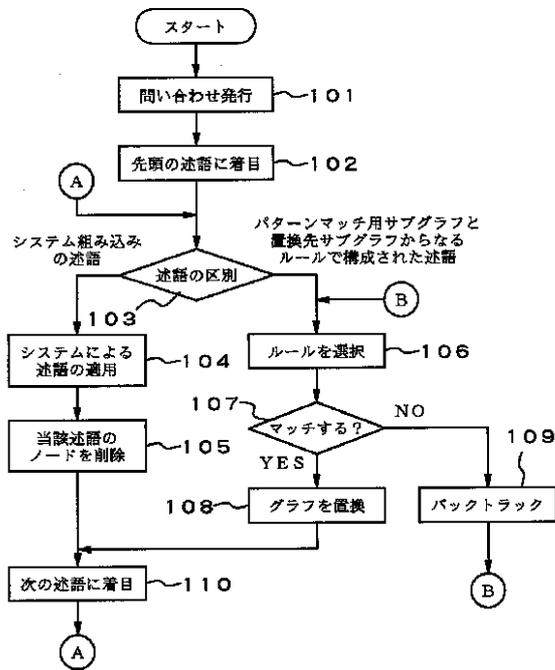
【図5】



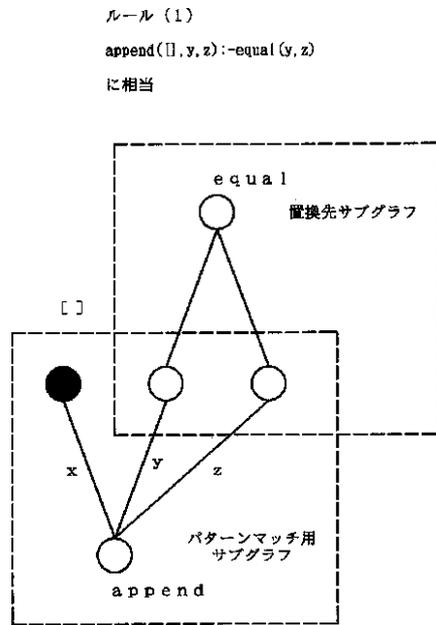
【図10】



【図1】

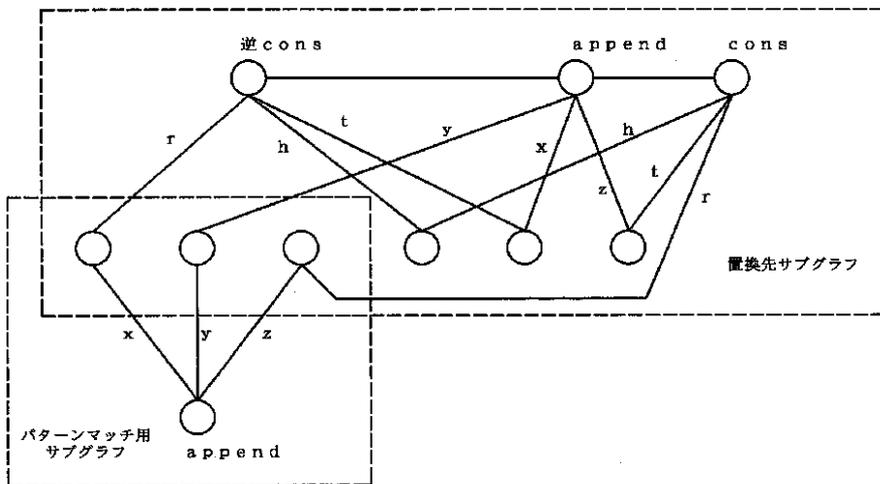


【図2】

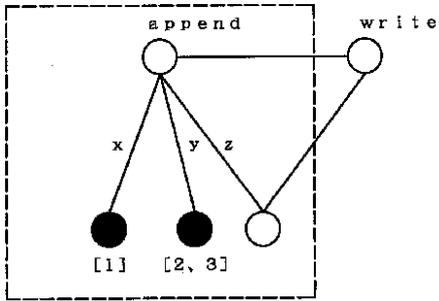


【図3】

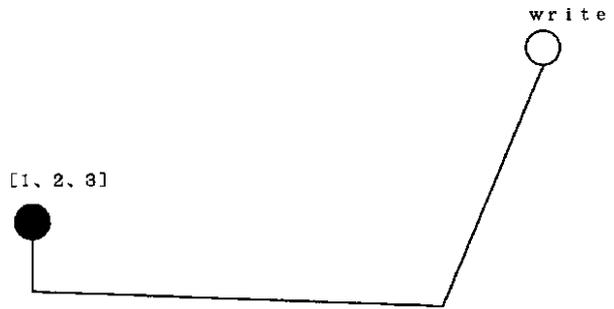
ルール (2) $append(x, y, z) :- cons(h, t, x), append(t, y, tl), cons(h, tl, z)$ に相当



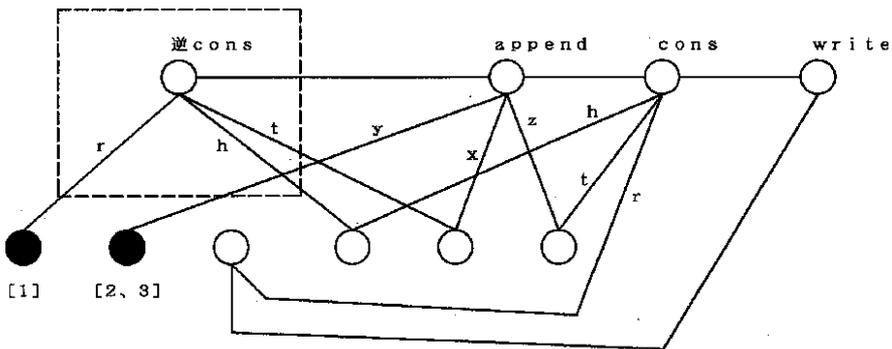
【図4】



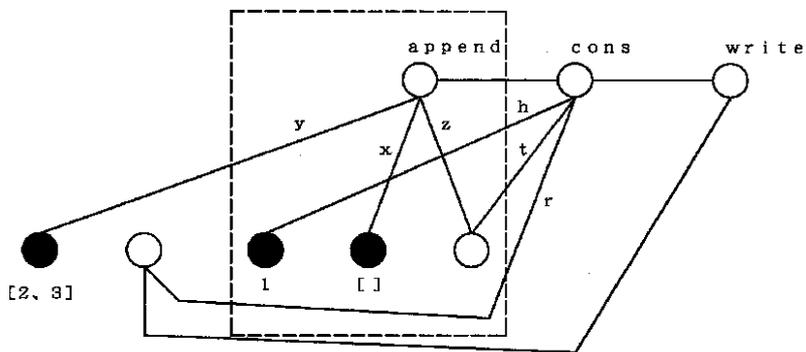
【図9】



【図6】



【図7】



【図8】

