(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6788249号
(P6788249)
(45)発行日　令和2年11月25日(2020.11.25)　(24)登録日　令和2年11月4日(2020.11.4)

(51)Int.Cl.　F I
G06N 5/04 (2006.01)　G06N 5/04

請求項の数 4　(全 16 頁)

(21)出願番号　特願2018-37413(P2018-37413)
(22)出願日　平成30年3月2日(2018.3.2)
(65)公開番号　特開2019-153047(P2019-153047A)
(43)公開日　令和1年9月12日(2019.9.12)
審査請求日　令和1年11月19日(2019.11.19)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(73)特許権者　504132272
国立大学法人京都大学
京都府京都市左京区吉田本町３６番地１
(74)代理人　100107766
弁理士　伊東　忠重
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100124844
弁理士　石原　隆治
(72)発明者　西野　正彬
東京都千代田区大手町一丁目５番１号　日本電信電話株式会社内
最終頁に続く

(54)【発明の名称】生成装置、生成方法及びプログラム

(57)【特許請求の範囲】

【請求項１】

帰納論理プログラミングの問題の解となる論理プログラムを表す仮説Σの各々を示す情報を生成する生成装置であって、

背景知識Ｂと、正例の訓練例を示す基礎原子論理式の集合ε^{+}と、前記仮説Σに含まれ得る縮小確定節の集合Ｐとが入力されると、前記集合ε^{+}に含まれる各基礎原子論理式Ｅについて、該基礎原子論理式Ｅが前記背景知識Ｂと仮説Σ_{1}との和集合の論理的帰結となるような仮説Σ_{1}に対応する論理関数ｆを表す第１の二分決定図を構築する第１の生成手段と、

前記背景知識Ｂと、負例の訓練例を示す基礎原子論理式の集合ε^{-}と、前記集合Ｐとが入力されると、前記集合ε^{-}に含まれる各基礎原子論理式Ｅについて、該基礎原子論理式Ｅが前記背景知識Ｂと仮説Σ_{2}との和集合の論理的帰結となるような仮説Σ_{2}に対応する論理関数ｇを表す第２の二分決定図を構築する第２の生成手段と、

前記第１の二分決定図の各々と、前記第２の二分決定図の各々とをＡｐｐｌｙ演算することで、第３の二分決定図を生成する第３の生成手段と、

を有し、

前記第１の生成手段及び第２の生成手段は、

前記基礎原子論理式Ｅを論理的帰結とする仮説の集合を表現する論理関数を［Ｅ］、前記集合Ｐに含まれる各縮小確定節$A \leftarrow B_{1}, \cdots, B_{n}$のうち、代入θに対してＥ＝Ａθを満たす縮小確定節$A \leftarrow B_{1}, \cdots, B_{n}$の添え字の集合を$J_{E}$として、

【数１４】

$$[E] = \bigvee_{i \in J_E} \left[x_i \wedge \left(\bigwedge_{j=1}^{n_i} [B_j \theta_i] \right) \right]$$

を再帰的にＡｐｐｌｙ演算することで、前記第１の二分決定図及び前記第２の二分決定図をそれぞれ構築する、ことを特徴とする生成装置。

【請求項２】

前記第３の生成手段は、

前記第１の二分決定図の各々を表すＭ個の論理関数をf_1，・・・f_M、前記第２の二分決定図の各々を表すＫ個の論理関数をg_1，・・・g_Kとして、

【数１５】

$$h = \left(\bigwedge_{i=1}^{M} f_i \right) \wedge \left(\bigwedge_{i=1}^{K} \neg g_i \right)$$

によって表される前記第３の二分決定図をＡｐｐｌｙ演算により生成する、ことを特徴とする請求項１に記載の生成装置。

【請求項３】

帰納論理プログラミングの問題の解となる論理プログラムを表す仮説Σの各々を示す情報を生成するコンピュータが、

背景知識Ｂと、正例の訓練例を示す基礎原子論理式の集合ε^{+}と、前記仮説Σに含まれ得る縮小確定節の集合Ｐとが入力されると、前記集合ε^{+}に含まれる各基礎原子論理式Ｅについて、該基礎原子論理式Ｅが前記背景知識Ｂと仮説Σ_1との和集合の論理的帰結となるような仮説Σ_1に対応する論理関数ｆを表す第１の二分決定図を構築する第１の生成手順と、

前記背景知識Ｂと、負例の訓練例を示す基礎原子論理式の集合ε^{-}と、前記集合Ｐとが入力されると、前記集合ε^{-}に含まれる各基礎原子論理式Ｅについて、該基礎原子論理式Ｅが前記背景知識Ｂと仮説Σ_2との和集合の論理的帰結となるような仮説Σ_2に対応する論理関数ｇを表す第２の二分決定図を構築する第２の生成手順と、

前記第１の二分決定図の各々と、前記第２の二分決定図の各々とをＡｐｐｌｙ演算することで、第３の二分決定図を生成する第３の生成手順と、

を実行し、

前記第１の生成手順及び第２の生成手順は、

前記基礎原子論理式Ｅを論理的帰結とする仮説の集合を表現する論理関数を［Ｅ］、前記集合Ｐに含まれる各縮小確定節$A \leftarrow B_1$，・・・，B_nのうち、代入θに対して$E = A\theta$を満たす縮小確定節$A \leftarrow B_1$，・・・，B_nの添え字の集合をJ_Eとして、

【数１６】

$$[E]=\bigvee_{i\in J_E}\left[x_i\wedge\left(\bigwedge_{j=1}^{n_i}[B_j\theta_i]\right)\right]$$

を再帰的にＡｐｐｌｙ演算することで、前記第１の二分決定図及び前記第２の二分決定図をそれぞれ構築する、ことを特徴とする生成方法。

【請求項４】

コンピュータを、請求項１又は２の何れか一項に記載の生成装置における各手段として機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、生成装置、生成方法及びプログラムに関する。

【背景技術】

【０００２】

論理プログラムを用いた機械学習手法である帰納論理プログラミング（ＩＬＰ：Inductive Logic Programming）が従来から知られている（非特許文献１）。帰納論理プログラミングでは、与えられた訓練データ（以降では、「訓練例」とも表す。）と整合性がとれるような論理プログラムを推定する技術である。このため、帰納論理プログラミングにより、論理プログラムを人手で設計しなくても、訓練例を与えるだけで、これらの訓練例と整合性がとれるような論理プログラムを獲得することができる。

【先行技術文献】

【非特許文献】

【０００３】

【非特許文献１】山本，「帰納論理プログラミングの基礎理論とその展開」，コンピュータソフトウェア，Vol.23，No. 2，pp. 29-44，2006.

【発明の概要】

【発明が解決しようとする課題】

【０００４】

ここで、従来の帰納論理プログラミングのアルゴリズムでは、与えられた訓練例と矛盾しない論理プログラム（解）を１つ獲得することを目的としている。しかしながら、実際には訓練例と矛盾しないような解が多数存在する可能性がある。

【０００５】

このため、従来の帰納論理プログラミングでは、与えられた訓練例と矛盾しない解を複数列挙することができず、例えば、与えられた訓練例と矛盾しない論理プログラムの中から所望の条件を満たす最適な論理プログラムを求めたい等の要求に応えることが困難であった。

【０００６】

本発明の実施の形態は、上記の点に鑑みてなされたもので、帰納論理プログラミングの全ての解を得ることを目的とする。

【課題を解決するための手段】

【０００７】

上記目的を達成するため、本発明の実施の形態は、帰納論理プログラミングの問題の解となる論理プログラムを表す仮説Σの各々を示す情報を生成する生成装置であって、背景知識Ｂと、正例の訓練例を示す基礎原子論理式の集合ε^{+}と、前記仮説Σに含まれ得る縮

小確定節の集合Ｐとが入力されると、前記集合ε^{+}に含まれる各基礎原子論理式Ｅについて、該基礎原子論理式Ｅが前記背景知識Ｂと仮説Σ_1との和集合の論理的帰結となるような仮説Σ_1に対応する論理関数ｆを表す第１の二分決定図を構築する第１の生成手段と、前記背景知識Ｂと、負例の訓練例を示す基礎原子論理式の集合ε^{-}と、前記集合Ｐとが入力されると、前記集合ε^{-}に含まれる各基礎原子論理式Ｅについて、該基礎原子論理式Ｅが前記背景知識Ｂと仮説Σ_2との和集合の論理的帰結となるような仮説Σ_2に対応する論理関数ｇを表す第２の二分決定図を構築する第２の生成手段と、前記第１の二分決定図の各々と、前記第２の二分決定図の各々とをＡｐｐｌｙ演算することで、第３の二分決定図を生成する第３の生成手段と、を有し、前記第１の生成手段及び第２の生成手段は、前記基礎原子論理式Ｅを論理的帰結とする仮説の集合を表現する論理関数を［Ｅ］、前記集合Ｐに含まれる各縮小確定節$A \leftarrow B_1, \cdots, B_n$のうち、代入θに対して$E = A\theta$を満たす縮小確定節$A \leftarrow B_1, \cdots, B_n$の添え字の集合を$J_E$として、所定の式を再帰的にＡｐｐｌｙ演算することで、前記第１の二分決定図及び前記第２の二分決定図をそれぞれ構築する、ことを特徴とする。

【発明の効果】

【０００８】

帰納論理プログラミングの全ての解を得ることができる。

【図面の簡単な説明】

【０００９】

【図１】二分決定図の一例を示す図である。

【図２】本発明の実施の形態における生成装置の機能構成の一例を示す図である。

【図３】本発明の実施の形態における生成処理部が実行する処理の流れの一例を示すフローチャートである。

【図４】本発明の実施の形態における生成装置のハードウェア構成の一例を示す図である。

【発明を実施するための形態】

【００１０】

以下、本発明の実施の形態について説明する。本発明の実施の形態では、帰納論理プログラミングにおいて、与えられた訓練例と整合性がとれる論理プログラムを全て得ることを目的とする。ただし、帰納論理プログラミングの解となる論理プログラムの数は指数的に増加する可能性がある。このため、これらの全ての論理プログラムを出力するようなシステムは現実的な設定では動作しない可能性がある。そこで、全ての論理プログラムを出力するのではなく、全ての解の集合を表す二分決定図（ＢＤＤ：Binary Decision Diagram）を構築することで、これら全ての論理プログラムの列挙を実現する。なお、ＢＤＤについては以下の参考文献１を参照されたい。

【００１１】

［参考文献１］

Bryant, R.E., "Graph-Based Algorithms for Boolean Function Manipulation", IEEE Trans. Computers, Vol.C-35, No.8, pp.677-691, 1986.

ＢＤＤは、論理プログラムの集合をグラフとして表現可能なデータ構造である。ＢＤＤを用いて論理プログラムの集合を表現することで、膨大な数の論理プログラムの集合を圧縮及び索引化することができるため、計算機の記憶媒体等に格納することができるようになる。また、ＢＤＤによって表現された論理プログラムの集合から、或る条件を満たす論理プログラムを１つ取り出すような操作も効率的に実行することが可能となる。

【００１２】

そこで、本発明の実施の形態では、与えられた訓練例と整合性がとれる全ての論理プログラムの集合の索引を表すＢＤＤを生成及び出力する生成装置１０について説明する。

【００１３】

まず、本発明の実施の形態に係る生成装置１０について説明する前に、いくつかの用語等について説明する。

【００１４】

＜一階述語論理＞

一階述語論理における項とは、（１）変数、（２）定数記号、及び（３）アリティｎの関数記号ｆとｎ個の項t_1，・・・，t_nとによって作られる記号列ｆ（t_1，・・・，t_n）のことである。

【００１５】

また、アリティｎの述語記号Ｐとｎ個の項t_1，・・・，t_nとによって作られる記号列Ｐ（t_1，・・・，t_n）を原子論理式という。原子論理式Ａ及びその否定￢Ａをリテラルといい、特にＡを「正リテラル」、￢Ａを「負リテラル」という。リテラルの換言（すなわち、「∨」）に現れる変数の全体を束縛して得られる論理式を節という。

【００１６】

一階述語論理における論理式の真偽値は解釈によって決定される。一階述語論理の言語Ｌに対する解釈Ｉは、空でない集合（ドメイン）Ｄによって決定される。すなわち、解釈Ｉによって、定数記号はＤの要素に割り当てられ、アリティｎの関数記号はD^nからＤへの写像に割り当てられ、アリティｎの述語記号はD^nから真偽値｛Ｔｒｕｅ，ｆａｌｓｅ｝への写像に割り当てられる。

【００１７】

論理式φが解釈Ｉの下で真である場合、解釈Ｉはφのモデルであるという。論理式の集合Σの任意のモデルが論理式φのモデルである場合、論理式φは、論理式の集合Σの論理的帰結であるといい、

【００１８】

【数１】

$$\Sigma \models \phi$$

と表す。一方で、論理式φが、論理式の集合Σの論理的帰結でない場合は、

【００１９】

【数２】

$$\Sigma \not\models \phi$$

と表す。

【００２０】

また、変数を含まない原子論理式を基礎原子論理式という。同様に、変数を含まないリテラルを基礎リテラルという。変数を含まない節を基礎節という。

【００２１】

一階述語論理において、変数x_1，・・・，x_mを項t_1，・・・，t_mに置き換えることを代入という。このような代入は、θ＝｛x_1／t_1，・・・，x_m／t_m｝と表される。また、一階述語論理の節Ｃに含まれる全ての変数x_1，・・・，x_mを項t_1，・・・，t_mに同時に置き換えたものをＣθと表す。

【００２２】

ここで、帰納論理プログラミングとは、節の有限集合

【００２３】
【数３】

$$\mathcal{B},\ \mathcal{E}^{+},\ \mathcal{E}^{-}$$

が与えられた場合に、全ての節Ｅ∈ε^{+}に対して、以下の式１が成り立ち、
【００２４】
【数４】

$$\Sigma \cup \mathcal{B} \models E \quad \cdots(\text{式}1)$$

、かつ、全ての節Ｅ∈ε^{-}に対して、以下の式２が成り立つような節の有限集合Σを求める問題のことである。なお、以降では、節の有限集合のことを仮説とも呼ぶ。
【００２５】
【数５】

$$\Sigma \cup \mathcal{B} \not\models E \quad \cdots(\text{式}2)$$

ここで、
【００２６】
【数６】

$$\mathcal{B}$$

は背景知識とも呼ばれる。この背景知識を、以降では、「背景知識Ｂ」とも表す。また、ε^{+}は正例、ε^{-}は負例とも呼ばれる。
【００２７】
例えば、
【００２８】

【数７】

$$\mathcal{B} = \emptyset.$$
$$\mathcal{E}^{+} = \{P(0), P(s(s(0))), P(s(s(s(s(0)))))\},$$
$$\mathcal{E}^{-} = \{P(s(0)), P(s(s(s(0))))\}.$$

が与えられたとすると、仮説Σ＝｛Ｐ（０）←，Ｐ（ｓ（ｓ（Ｘ）））←Ｐ（ｘ）｝は、この機能論理プログラミングの問題の１つの解である。なお、Ｐは述語記号、ｓは関数記号である。また、Ｐ（０）←、Ｐ（ｓ（ｓ（Ｘ）））←Ｐ（ｘ）は確定節を表す。

【００２９】

確定節とは、正リテラルをちょうど１つ含む節のことである。確定節は、原子論理式Ａ，$Ｂ_1$，・・・，$Ｂ_n$を用いて、Ａ←$Ｂ_1$，・・・，$Ｂ_n$又はＡ←と表される。なお、Ａ←の形で表される確定節は単位節と呼ばれる。また、変数を含まない確定節を基礎確定節といい、変数を含まない単位節を基礎単位節という。

【００３０】

更に、或る原子論理式Ａに含まれる関数記号、定数記号及び変数記号の総数を｜Ａ｜と表す。任意の代入θ及びｉ＝１，・・・，ｎに対して、｜Ａθ｜≧｜$Ｂ_i$θ｜を満たす場合、確定節Ａ←$Ｂ_1$，・・・，$Ｂ_n$を縮小確定節という。

【００３１】

＜二分決定図＞

二分決定図（ＢＤＤ）は、論理関数を有向非巡回グラフとして表現するデータ構造である。一例として、論理関数（$ｘ_1$∧$ｘ_2$）∨（$ｘ_1$∧$ｘ_3$）∨（$ｘ_2$∧$ｘ_3$）を表現するＢＤＤを図１に示す。

【００３２】

ＢＤＤは、終端ノードと分岐ノードとの２種類のノードを有する。終端ノードは、当該ノードを始点とする枝（アーク）を持たないノードである。図１に示す例では、終端ノードは四角形で表されたノード（ノードｎ５及びノードｎ６）である。終端ノードには、⊥終端ノード（以降、「第１終端ノード」という。）と、

【００３３】

【数８】

⊤終端ノード

（以下、「第２終端ノード」という。）との２種類のノードがある。１つのＢＤＤには、第１終端ノードと第２終端ノードとがそれぞれ高々１つずつ存在する。

【００３４】

一方で、分岐ノードとは、終端ノードではないノードのことである。図１に示す例では、分岐ノードは円形で表されたノード（ノードｎ１、ノードｎ２、ノードｎ３及びノードｎ４）である。各分岐ノードには、ＢＤＤによって表現される論理関数の変数全体の集合のうち、当該分岐ノードに対応する要素を表すラベルが付与される。図１に示す例では、ノードｎ１には、変数$ｘ_1$を表す要素１がラベルとして付与されている。同様に、ノード

ｎ２及びノードｎ３には、変数x_2を表す要素２がラベルとして付与されている。同様に、ノードｎ４には、変数x_3を表す要素３がラベルとして付与されている。

【００３５】

また、各分岐ノードには、当該分岐ノードを始点とする枝が必ず２つ存在し、それぞれｌｏ枝及びｈｉ枝と呼ばれる。図１に示す例では、ｌｏ枝は破線、ｈｉ枝は実線で表されており、ノードｎ１のｌｏ枝はノードｎ２を指し、ｈｉ枝はノードｎ３を指している。この場合、ノードｎ１は親ノード、ノードｎ２及びノードｎ３は子ノードとなる。

【００３６】

同様に、ノードｎ２のｌｏ枝はノードｎ５を指し、ｈｉ枝はノードｎ４を指している。この場合、ノードｎ２は親ノード、ノードｎ５及びノードｎ４は子ノードとなる。他のノードも同様に、ノードｎ３のｌｏ枝及びｈｉ枝はそれぞれノードｎ４及びノードｎ６を指し、ノードｎ３が親ノード、ノードｎ４及びノードｎ６が子ノードとなる。ノードｎ４のｌｏ枝及びｈｉ枝はそれぞれノードｎ５及びノードｎ６を指し、ノードｎ４が親ノード、ノードｎ５及びノードｎ６が子ノードとなる。

【００３７】

ＢＤＤには親ノードを持たないノードが必ず１つのみ存在し、根ノードと呼ばれる。図１に示す例では、ノードｎ１が根ノードである。

【００３８】

根ノードから第２終端ノードまでの各経路は、ＢＤＤが表現する論理関数を真とするような変数の割り当てに対応している。ＢＤＤの各経路を３つの変数（x_1，x_2，x_3）で表現することとし、各変数の値は０又は１を取るものとし、ｈｉ枝を通る経路のときは変数の値が１であり、ｌｏ枝を通る経路のときは変数の値が０であるものとすれば、図１に示すＢＤＤにおいて、根ノードから第２終端ノードに至る各経路に対応する変数の割り当ては値（１，１，＊）、（１，０，１）、（０，１，１）で表現できる。ここで、＊は０又は１のいずれかの値を割り当てることに相当する。

【００３９】

例えば、（１，１，＊）は、図１に示すＢＤＤにおいて、ノードｎ１、ノードｎ３及びノードｎ６を辿る経路を表す。ここで、この場合、変数x_3を表す要素３のラベルが付与されているノードを辿らないが、これは、図１に示すＢＤＤでは冗長な節点が削除されているためである（すなわち、図１に示すＢＤＤは既約である。）。

【００４０】

同様に、（１，０，１）は、図１に示すＢＤＤにおいて、ノードｎ１、ノードｎ３、ノードｎ４及びノードｎ６を辿る経路を表す。同様に、（０，１，１）は、図１に示すＢＤＤにおいて、ノードｎ１、ノードｎ２、ノードｎ４及びノードｎ６を辿る経路を表す。

【００４１】

なお、根ノードから第２終端ノード（及び第１終端ノード）までの各経路が表現する変数の順序が同じであるＢＤＤを「順序付きＢＤＤ」ともいう。図１に示すＢＤＤは順序付きＢＤＤ（より正確には、既約な順序付きＢＤＤ）である。本発明の実施の形態において、ＢＤＤは、順序付きＢＤＤ（及び既約な順序付きＢＤＤ）であるものとする。

【００４２】

ＢＤＤの各ノードは、当該ノードを根ノードとする部分ＢＤＤを表現しており、部分ＢＤＤは１つの論理関数を表現している。図１に示す例では、例えば、ノードｎ２を根ノードとする部分ＢＤＤは、変数x_2及びx_3に対して定義される部分論理関数$x_2 \wedge x_3$を表現している。なお、第１終端ノードはＢＤＤによって表現される論理関数の真偽値が０であることに対応し、第２終端ノードはＢＤＤによって表現される論理関数の真偽値が１であることに対応するものとする。

【００４３】

なお、以降では、ＢＤＤのノードはＢ個存在するものとし、各ノードをb_1，・・・，b_Bと表す。また、b_1を根ノード、b_Bを第２終端ノードとし、任意の２つのノードb_i及びb_jに対して、$i<j$ならばb_iはb_jの子ノードにはなり得ないものとする。

【００４４】
　ＢＤＤは、各ノードについて、（ノードのＩＤ，ラベル，ｈｉ枝の指すノードのＩＤ，ｌｏ枝の指すノードのＩＤ）の４つの組で表現できる。図１に示すＢＤＤは６つノードを持つため、
【００４５】
【数９】

$$[(n1, 1, n3, n2), (n2, 2, n4, n5), (n3, 2, n6, n4), (n4, 3, n6, n5), (n5, \perp, -, -), (n6, \top, -, -)]$$

と表現することができる。
【００４６】
　＜Ａｐｐｌｙ演算＞
　論理関数ｆを表現するＢＤＤをＺ（ｆ）、論理関数ｇを表現するＢＤＤをＺ（ｇ）として、論理関数ｆとｇとの間に二項演算を適用することによって得られる論理関数ｆ∨ｇ及びｆ∧ｇをそれぞれ表現するＢＤＤをＺ（ｆ∨ｇ）及びＺ（ｆ∧ｇ）と表すものとする。Ｚ（ｆ）及びＺ（ｇ）からＺ（ｆ∨ｇ）やＺ（ｆ∧ｇ）を求める演算は、Ａｐｐｌｙ演算と呼ばれる。Ａｐｐｌｙ演算は、Ｚ（ｆ）及びＺ（ｇ）の大きさに比例する演算時間で実行できることが知られている。なお、Ａｐｐｌｙ演算については上記の参考文献１を参照されたい。
【００４７】
　＜生成装置１０＞
　次に、本発明の実施の形態に係る生成装置１０について説明する。
【００４８】
　　≪機能構成≫
　まず、本発明の実施の形態における生成装置１０の機能構成について、図２を参照しながら説明する。図２は、本発明の実施の形態における生成装置１０の機能構成の一例を示す図である。
【００４９】
　図２に示すように、本発明の実施の形態における生成装置１０は、生成処理部１００を有する。生成処理部１００は、生成装置１０にインストールされた１以上のプログラムがＣＰＵ（Central Processing Unit）等に実行させる処理により実現される。
【００５０】
　生成処理部１００は、背景知識Ｂと、訓練例の集合ε^{+}及びε^{-}と、仮説Σに含まれ得る節の集合Ｐとを入力する。ここで、訓練例は全て基礎単位節であり、背景知識Ｂ、集合ε^{+}、集合ε^{-}及び集合Ｐは有限集合であるものとする。また、集合Ｐに含まれる節は全て縮小確定節であり、集合Ｐに含まれる縮小確定節の数をＮとする。なお、集合ε^{+}は正例の訓練例の集合であり、集合ε^{-}は負例の訓練例の集合である。
【００５１】
　このとき、生成処理部１００は、入力された集合ε^{+}及びε^{-}と背景知識Ｂとに基づいて、上記の式１及び式２が成立するような仮説Σの集合を求める。なお、仮説Σは集合Ｐの部分集合である。
【００５２】
　本発明の実施の形態では、仮説ΣをＮ次元の２値ベクトルｘを用いて表現する。２値ベクトルｘのｉ番目の成分をx_iとする。そして、集合Ｐに含まれるｉ番目の縮小確定節が仮説Σに含まれる場合はx_i＝１、ｉ番目の縮小確定節が仮説Σに含まれない場合はx_i

＝０とする。このＮ次元２値ベクトルｘによって仮説Σを表現することで、仮説Σの集合は論理関数として表現できる。すなわち、仮説Σを表す２値ベクトルｘが帰納論理プログラミングの問題の解である場合はｆ（ｘ）＝１、解でない場合はｆ（ｘ）＝０となるような論理関数ｆは、全ての解の集合を表現していると見做すことができる。

【００５３】

したがって、この２値ベクトルｘを構成する各変数x_iを、ＢＤＤを表現する変数とみれば（すなわち、ＢＤＤの各ノードに付与されるｉ個のラベルがそれぞれ各変数x_iを表すとすれば）、論理関数ｆを表現するＢＤＤによって、仮説Σの集合を表現することができる。このとき、ＢＤＤの根ノードから第２終端ノードまでに至る経路の各々が、帰納論理プログラミングの問題の解となる仮説を表す。一方で、ＢＤＤの根ノードから第１終端ノードまでに至る経路の各々が、帰納論理プログラミングの問題の解とならない仮説を表す。

【００５４】

このことは、ＢＤＤの各経路に対応する変数の割り当ては、当該経路に対応する仮説の索引と捉えることができることを意味する。

【００５５】

すなわち、生成処理部１００は、入力された集合ε^+及びε^-と背景知識Ｂとに基づいて、ＢＤＤ　Ｚ（ｈ）を構築する。そして、生成処理部１００は、構築したＢＤＤ　Ｚ（ｈ）を出力する。このＺ（ｈ）の根ノードから第２終端ノードまでに至る経路に対応する変数の割り当てが、帰納論理プログラミングの解である仮説Σの索引である。したがって、生成処理部１００は、入力された集合ε^+及びε^-と背景知識Ｂとに基づいて、ＢＤＤ　Ｚ（ｈ）を構築することにより、帰納論理プログラミングの解である仮説Σの索引を生成すると言うこともできる。

【００５６】

なお、生成処理部１００は、帰納論理プログラミングの解である仮説Σの索引として、構築したＢＤＤ　Ｚ（ｈ）を出力しても良いし、このＢＤＤ　Ｚ（ｈ）の根ノードから第２終端ノードまでに至る経路に対応する変数の割り当ての集合を出力しても良い。

【００５７】

ここで、生成処理部１００には、第１仮説集合生成部１０１と、第２仮説集合生成部１０２と、第３仮説集合生成部１０３とが含まれる。

【００５８】

第１仮説集合生成部１０１は、背景知識Ｂと、訓練例の集合ε^+と、集合Ｐとを入力して、全ての基礎原子論理式$E \in \varepsilon^+$について、上記の式１を満たす仮説Σ_1の集合に対応する論理関数を表現するＢＤＤを構築する。すなわち、ε^+に基礎原子論理式がＭ個含まれるとした場合、ｉ（ｉ＝１，・・・，Ｍ）毎に、第１仮説集合生成部１０１は、ｉ番目の基礎原子論理式E_iに対して、

【００５９】

【数１０】

$$\mathcal{B} \cup \Sigma_1 \models E_i$$

となるような仮説Σ_1の集合を表現する論理関数f_iを表すＢＤＤ　Ｚ（f_i）を構築する。ＢＤＤの構築方法については後述する。なお、ε^+に含まれる訓練例は全て基礎単位節であることから、これらは全て基礎原子論理式である。

【００６０】

第２仮説集合生成部１０２は、背景知識Ｂと、訓練例の集合ε^-と、集合Ｐとを入力して、全ての原子論理式$E \in \varepsilon^-$について、上記の式１を満たす仮説Σ_2の集合に対応する論理関数を表現するＢＤＤを構築する。すなわち、ε^-に基礎原子論理式がＫ個含まれるとした場合、ｉ（ｉ＝１，・・・，Ｋ）毎に、第２仮説集合生成部１０２は、ｉ番目の基礎原子論理式E_iに対して、

【００６１】

【数１１】

$$\mathcal{B} \cup \Sigma_2 \models E_i$$

となるような仮説Σ_2の集合を表現する論理関数g_iを表すＢＤＤ　Ｚ（g_i）を構築する。ＢＤＤの構築方法については後述する。なお、ε^-に含まれる訓練例は全て基礎単位節であることから、これらは全て基礎原子論理式である。

【００６２】

ここで、上記では、負例である訓練例$E_i \in \varepsilon^-$に対しても上記の式１を満たすΣ_2の集合に対応する論理関数g_iを表すＢＤＤ　Ｚ（g_i）を構築した。これは、後述する式３において、各論理関数g_iに対して否定を示す論理記号（¬）を付与するためである。ただし、後述する式３において、各論理関数g_iに対して否定を示す論理記号（¬）を付与しない場合には、第２仮説集合生成部１０２は、上記の式２を満たす仮説Σ_2の集合に対応する論理関数を表現するＢＤＤを構築しても良い。

【００６３】

第３仮説集合生成部１０３は、第１仮説集合生成部１０１で構築したＺ（f_i）と、第２仮説集合生成部１０２で構築したＺ（g_i）とを入力して、これらのＢＤＤのＡｐｐｌｙ演算を行うことにより、以下の式３に示す論理式ｈを表すＢＤＤ　Ｚ（ｈ）を構築する。

【００６４】

【数１２】

$$h = \left(\bigwedge_{i=1}^{M} f_i\right) \wedge \left(\bigwedge_{i=1}^{K} \neg g_i\right) \quad \cdots(\text{式}3)$$

≪処理の流れ≫

次に、本発明の実施の形態における生成処理部１００が実行する処理の流れについて、図３を参照しながら説明する。図３は、本発明の実施の形態における生成処理部１００が実行する処理の流れの一例を示すフローチャートである。なお、以降では、背景知識Ｂと、訓練例の集合ε^+及びε^-と、集合Ｐとが生成処理部１００に入力されたものとする。

【００６５】

ステップＳ１０１：生成処理部１００の第１仮説集合生成部１０１は、背景知識Ｂと、正例の訓練例の集合ε^+と、集合Ｐとを入力して、ｉ＝１，・・・，Ｍに対して、上述したＺ（f_i）を構築する。

【００６６】

ここで、ＢＢＤ　Ｚ（f_i）の構築方法について説明する。以降では、f_iの添字ｉを固定して、単に「ｆ」と表し、Ｚ（ｆ）の構築方法について説明する。

【００６７】

或る基礎原子論理式Ｅを論理的帰結とするような仮説の集合を表現する論理関数を［Ｅ］とする。また、集合Ｐに含まれる各縮小確定節に対して添字ｉ（ｉ＝１，・・・，Ｎ）が付与されているものとして、集合Ｐに含まれる各縮小確定節Ａ←B_1，・・・，B_nのうち、或る代入θに対してＥ＝Ａθを満たすような縮小確定節の添字の集合をJ_Eとする。すると、［Ｅ］は、

【００６８】

【数１３】

$$[E] = \bigvee_{i \in J_E} \left[x_i \wedge \left(\bigwedge_{j=1}^{n_i} [B_j \theta_i] \right) \right] \quad \cdots(\text{式}4)$$

となる。ここで、θ_iはｉ番目の縮小確定節に対してＥ＝Ａθ_iとなるような代入を表す。このような代入は各縮小確定節に対して一意に決定される。また、n_iはｉ番目の縮小確定節を表す原子論理式B_1，・・・，B_nの数である。ｉ番目の縮小確定節は、原子論理式Ａ，B_1，・・・，B_{ni}を用いて、Ａ←B_1，・・・，B_{ni}と表される（ただし、「ｎｉ」は、実際には「ｉ」を下付きで表した「n_i」である。）。

【００６９】

上記の式４は、論理関数［Ｅ］が論理関数［$B_j\theta_i$］により再帰的に求められることを表している。縮小確定節の性質により、$B_j\theta_i$は常に基礎原子論理式となり、かつ、その大きさ｜$B_j\theta_i$｜は常に｜Ｅ｜より小さくなる。他方で、背景知識Ｂと、正例の集合ε^+と、集合Ｐとはいずれも有限集合であるため、これらの集合に含まれる関数記号、定数記号、述語記号も有限である。このため、これらの関数記号、定数記号、述語記号によって構成される基礎原子論理式のうち、ε^+に含まれる最大の基礎原子論理式よりも小さいものの集合も常に有限となる。このような大きさが或る値以下の各基礎原子論理式Ｅについて、論理関数［Ｅ］を表すＢＤＤを、上記の式４により再帰的にＡｐｐｌｙ演算を実行して構築していくことによって、ε^+に含まれる各原子論理式Ｅについて論理関数ｆ＝［Ｅ］を表すＢＤＤ　Ｚ（ｆ）を構築することができる。

【００７０】

ステップＳ１０２：生成処理部１００の第２仮説集合生成部１０２は、背景知識Ｂと、負例の訓練例の集合ε^-と、集合Ｐとを入力して、ｉ＝１，・・・，Ｋに対して、上述したＺ（g_i）を構築する。

【００７１】

ここで、ＢＢＤ　Ｚ（g_i）の構築方法について説明する。以降では、g_iの添字ｉを固定して、単に「ｇ」と表し、Ｚ（ｇ）の構築方法について説明する。なお、Ｚ（ｇ）の構築方法は、上述したＺ（ｆ）の構築方法と同様であるため、簡略化して説明する。

【００７２】

或る基礎原子論理式Ｅを論理的帰結とするような仮説の集合を表現する論理関数を［Ｅ］とする。また、集合Ｐに含まれる各縮小確定節Ａ←B_1，・・・，B_nのうち、或る代入θに対してＥ＝Ａθを満たすような縮小確定節の添字の集合をJ_Eとする。すると、［Ｅ］は、上記の式４となる。

【００７３】

また、負例の集合ε^-も有限集合であるため、背景知識Ｂ、負例の集合ε^-及び集合Ｐに含まれる関数記号、定数記号、述語記号によって構成される基礎原子論理式のうち、ε

$^{-}$に含まれる最大の基礎原子論理式よりも小さいものの集合も常に有限となる。したがって、このような大きさが或る値以下の各基礎原子論理式Ｅについて、論理関数［Ｅ］を表すＢＤＤを、上記の式４により再帰的にＡｐｐｌｙ演算を実行して構築していくことによって、ε^{-}に含まれる各原子論理式Ｅについて論理関数ｇ＝［Ｅ］を表すＢＤＤ　Ｚ（ｇ）を構築することができる。

【００７４】

ステップＳ１０３：生成処理部１００の第３仮説集合生成部１０３は、第１仮説集合生成部１０１で構築したＺ（f_i）と、第２仮説集合生成部１０２で構築したＺ（g_i）とを入力して、これらのＢＤＤのＡｐｐｌｙ演算を行うことにより、上記の式３に示す論理式ｈを表すＢＤＤ　Ｚ（ｈ）を構築する。

【００７５】

これにより、生成処理部１００により、帰納論理プログラミングの解である仮説Σの索引として、Ｚ（ｈ）又は当該Ｚ（ｈ）の根ノードから第２終端ノードまでに至る経路に対応する変数の割り当ての集合が出力される。

【００７６】

以上のように、本発明の実施の形態における生成装置１０では、帰納論理プログラミングの解である仮説Σの索引の集合を、ＢＤＤ又は当該ＢＤＤの根ノードから第２終端ノードまでに至る経路に対応する変数の割り当ての集合として出力することができる。このため、本発明の実施の形態における生成装置１０によれば、膨大な数になる可能性がある帰納論理プログラミングの解の集合をＢＤＤとして圧縮・索引化して全列挙することができるようになる。このように全ての解の集合をＢＤＤとして索引化して表現することで、例えば、これらの解の中から所望の条件を満たす解を選び出して利用する等といった事も容易に行うことができるようになる。

【００７７】

≪ハードウェア構成≫

最後に、本発明の実施の形態における生成装置１０のハードウェア構成について、図４を参照しながら説明する。図４は、本発明の実施の形態における生成装置１０のハードウェア構成の一例を示す図である。

【００７８】

図４に示すように、本発明の実施の形態における生成装置１０は、入力装置２０１と、表示装置２０２と、外部Ｉ／Ｆ２０３と、ＲＡＭ（Random Access Memory）２０４と、ＲＯＭ（Read Only Memory）２０５と、ＣＰＵ２０６と、通信Ｉ／Ｆ２０７と、補助記憶装置２０８とを有する。これら各ハードウェアは、それぞれがバスＢを介して通信可能に接続されている。

【００７９】

入力装置２０１は、例えばキーボードやマウス、タッチパネル等であり、ユーザが各種操作を入力するのに用いられる。表示装置２０２は、例えばディスプレイ等であり、生成装置１０の処理結果を表示する。なお、生成装置１０は、入力装置２０１及び表示装置２０２の少なくとも一方を有していなくても良い。

【００８０】

外部Ｉ／Ｆ２０３は、外部装置とのインタフェースである。外部装置には、記録媒体２０３ａ等がある。生成装置１０は、外部Ｉ／Ｆ２０３を介して、記録媒体２０３ａ等の読み取りや書き込みを行うことができる。記録媒体２０３ａには、生成装置１０が有する各機能を実現するプログラム等が記録されていても良い。

【００８１】

記録媒体２０３ａには、例えば、フレキシブルディスク、ＣＤ（Compact Disc）、ＤＶＤ（Digital Versatile Disk）、ＳＤメモリカード（Secure Digital memory card）、ＵＳＢ（Universal Serial Bus）メモリカード等がある。

【００８２】

ＲＡＭ２０４は、プログラムやデータを一時保持する揮発性の半導体メモリである。Ｒ

ＯＭ２０５は、電源を切ってもプログラムやデータを保持することができる不揮発性の半導体メモリである。ＲＯＭ２０５には、例えば、ＯＳ（Operating System）設定やネットワーク設定等が格納されている。

【００８３】

ＣＰＵ２０６は、ＲＯＭ２０５や補助記憶装置２０８等からプログラムやデータをＲＡＭ２０４上に読み出して処理を実行する演算装置である。

【００８４】

通信Ｉ／Ｆ２０７は、生成装置１０を通信ネットワークに接続するためのインタフェースである。生成装置１０が有する各機能を実現するプログラムは、通信Ｉ／Ｆ２０７を介して、所定のサーバ装置等から取得（ダウンロード）されても良い。

【００８５】

補助記憶装置２０８は、例えばＨＤＤ（Hard Disk Drive）やＳＳＤ（Solid State Drive）等であり、プログラムやデータを格納している不揮発性の記憶装置である。補助記憶装置２０８に格納されているプログラムやデータには、例えば、ＯＳ、当該ＯＳ上において各種機能を実現するアプリケーションプログラム、生成装置１０が有する各機能を実現するプログラム等がある。

【００８６】

本発明の実施の形態における生成装置１０は、図４に示すハードウェア構成を有することにより、上述した各種処理を実現することができる。なお、図４では、本発明の実施の形態における生成装置１０が１台の装置で実現される場合について説明したが、これ限られない。本発明の実施の形態における生成装置１０は、複数台の装置で実現されていても良い。

【００８７】

本発明は、具体的に開示された上記の実施形態に限定されるものではなく、特許請求の範囲から逸脱することなく、種々の変形や変更が可能である。

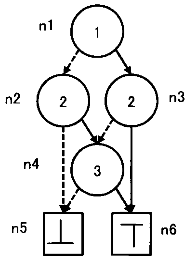
【符号の説明】

【００８８】

１０　　　生成装置
１０１　　第１仮説集合生成部
１０２　　第２仮説集合生成部
１０３　　第３仮説集合生成部

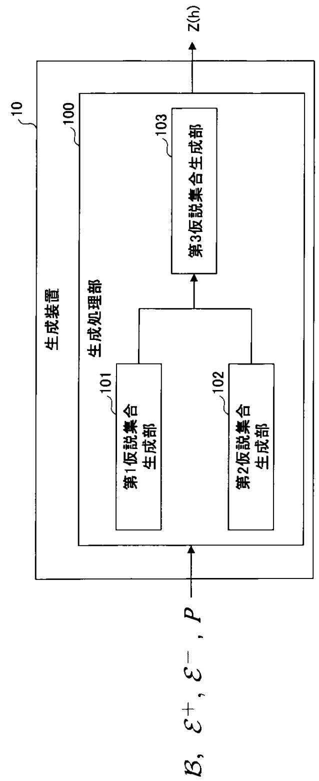
【図１】

二分決定図の一例を示す図



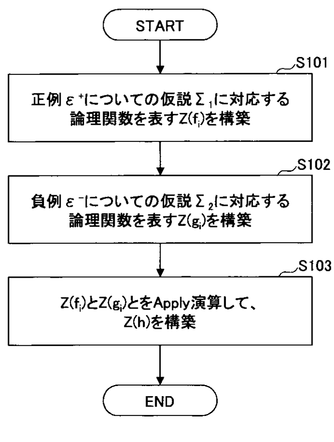
【図２】

本発明の実施の形態における生成装置の機能構成の一例を示す図



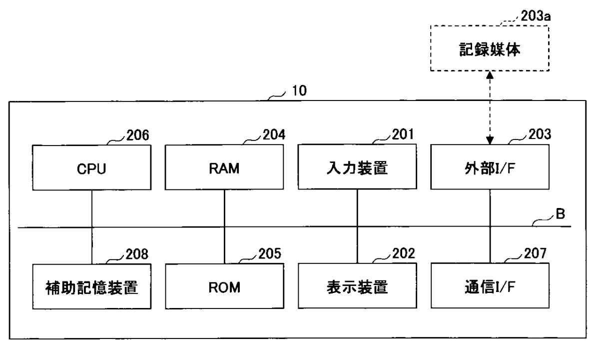
【図３】

本発明の実施の形態における
生成処理部が実行する処理の流れの一例を示すフローチャート



【図４】

本発明の実施の形態における生成装置のハードウェア構成の一例を示す図



フロントページの続き

(72)発明者　山本　章博
京都府京都市左京区吉田本町３６番地１　国立大学法人京都大学内
(72)発明者　新藤　光
京都府京都市左京区吉田本町３６番地１　国立大学法人京都大学内

審査官　多賀　実

(56)参考文献　特開２０１４－１６０４２３（ＪＰ，Ａ）
山口 慧 外2名，「二分決定グラフを用いた命題論理式の前提の列挙」，第９３回人工知能基本問題研究会資料（ＳＩＧ－ＦＰＡＩ－Ｂ３０４），一般社団法人 人工知能学会，２０１４年 3月　3日，pp.119-126

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｎ　５／００－５／０４
Ｇ０６Ｆ　８／００－８／７７
Ｇ０６Ｆ　９／４４－９／４５５