

推論過程からの概念学習(2) —概念構造の構成要因—

CLIP: Concept Learning from Inference Pattern (2) —Concept Hierarchy Formation—

吉田 健一* 元田 浩*
Ken'ichi Yoshida Hiroshi Motoda

* 日立製作所基礎研究所
Advanced Research Laboratory, Hitachi Ltd., Hatoyama, Saitama 350-03, Japan.

1991年7月1日 受理

Keywords: hierarchical understanding, abstraction, knowledge representation, machine learning.

Summary

We view a set of chunks, the use of which makes inference more efficient, as a concept. The idea is based on the assumption that a chunk that appears often in an inference may mean something important. The extraction of a chunk is solely based on finding the repetition of a typical inference pattern in a given environment. This idea, implemented as CLIP (Concept Learning from Inference Pattern), adapts Genetic Algorithm like parallel search algorithm and when applied to the digraphs of a carry chain circuit, CLIP extracted the chunks corresponding to analog NOR and NOT.

This paper discusses some of the important factors for concept hierarchy formation. Introduction of approximation is very important to step up to a more abstract level concept. This can also be processed as a reduction of digraph. Another important factor for the concept hierarchy formation is the characteristics of the inference system. This must be reflected on the matching cost. The different weight for the matching cost generates a hierarchy of different levels/depths. Environment of the inference system is also important. It must be reflected on the choice of color. Choice of a different color forms a hierarchy of different kinds. Presence of noise effects the performance, but the analysis indicates that CLIP can cope with a certain type of noise.

1. ま え が き

推論過程の分析結果から推論を効率的に行うという点で有効な概念を生成・学習する概念学習の方法 (CLIP: Concept Learning from Inference Pattern) を提案した⁽¹⁾。

従来概念学習といえば、対象がある概念を満足するか否かの分類木を正例・負例より学習する研究が多く、学習した概念の利用まで含めて考えた研究例は少なかった。前報⁽¹⁾では「概念」を「利用することにより推論が簡単になる言葉」であると捉え、推論過程からの概念学習を提案した。この学習方法によれば、例えば電気回路を対象とした電圧・電流の変化に関する定性

推論の過程を分析し、推論過程を簡単にする単語として、NOR/NOT に相当する単語を自動的に生成・学習できる。あらかじめ概念を形成するための特別の Heuristics を入力しておかなくても「推論を簡単にする」単語として抽象度の高い概念が生成される点の特徴である。

本報では「本当に CLIP が概念生成として意味があるのか？」および「概念構造に影響を与える要因としてどのようなものがあるのか」の2点を考察し、さらに関連研究との比較を試み、今後の研究課題についても言及する。

2. 概念構造の自動生成に関する考察

2・1 概念生成と概念抽出

本節では「本当にこれが概念生成として意味があるのか？」および「生成した概念により、本当に推論を高速化できるのか？」の2点を説明する。

CLIPは例題からの学習方法ではなく、学習・推論システムが置かれた環境からの学習方法である⁽¹⁾。別の言い方をすると、CLIPは環境に含まれている固まりを抽出しているだけである。したがって「CLIPが概念生成の能力を持っているか？」という問いは、「環境に含まれている固まりを見つける能力を概念生成能力と認めるか？」という問いと等価である。

著者らはこれを「見つけた固まりにより何らかの推論が簡単になる」という条件のもとで「Yes」と考えている。著者らは人間の持つ概念、例えば物理学にしても、人間が環境の変化の説明を容易にするために、もともと環境が持っていた固まりを抽出したものすぎないと思っている。したがって、人間も未知の概念を形成するときにはCLIPと類似の操作を行っていると考えている。人間はデータに重要度をつけて判別し、ノイズの排除が行える、初めに与えられた環境の外側まで外挿する能力がある等、明らかにCLIPの機能に含まれない能力も持っているが、それを考慮したうえでもCLIPの機能は人間の概念形成能力の重要な側面はカバーしているのではないかと考えている。

以上より2番目の質問への回答が重要となる。これは直観的に明らかと思われるが少し考察する。Fig. 1に前報⁽¹⁾の実験結果として得られたグラフの書換え

規則 (Macro Rule) および推論規則 (Interpretation Rule) を示す。また、この結果を得るのに使用したグラフの大きさの評価式は

グラフの大きさ＝

点の数＋辺の数

+ \sum 推論規則の大きさ 1

+ \sum 推論規則の大きさ 2

－Patternによるグラフの書換えが

起こった回数

(1)

であった。式(1)において第1項は、推論システムの記憶能力に対する負荷、第2項は、パターンマッチ能力に対する負荷、第3、4項は、記憶能力・パターンマッチ能力両者に対する負荷に対応している⁽¹⁾。CLIPでは、推論対象はEntity-Relationship Modelで表され、推論は特定のデータ間の関係の利用方法を記述した推論規則によりデータの値が順次求められることで実行されることを前提としている。推論対象のEntity-Relationship Modelが入力された段階でFig. 1に例示したグラフの書換え規則を、複数のRelationの組合せがあるときに新しいRelationに書き直すための規則とみなしModelを書き換え、やはりFig. 1に例示した推論規則を用いれば、得られる推論過程は小さなグラフに相当する負荷の小さなものとなる。

実際にはModelの書換えにも負荷がかかる。また、推論システム特性により考慮すべき項目および項目ごとの重要度は変化するが、評価式の係数を変えることにより、このような推論システム特性を考慮することは可能である。実験に用いたプログラムは逆ポーランド記法に準拠した簡易インタプリタを備えており、グラフの大きさの評価式を定義できるようになっ

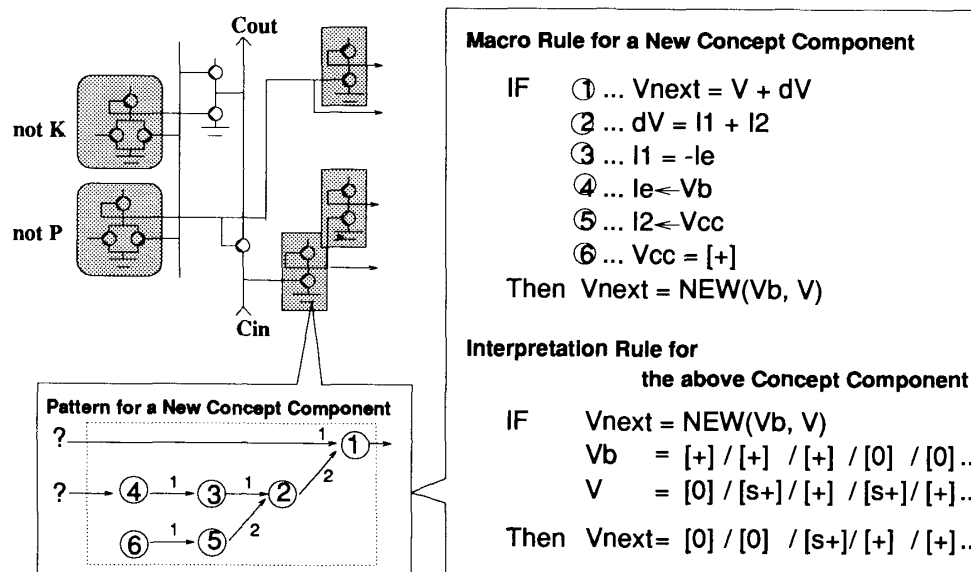


Fig. 1 Resulting rule.

ている。2・3 節で Model の書換えの負荷を考慮した評価式を用いた実験結果を示す。

2・2 近似と概念学習

Fig. 1 に示した結果を分析すると、EBL⁽²⁾⁽³⁾ の Macro Rule に相当する書換え規則は NOR や NOT 回路に相当する回路方程式の固まりを判別できるものを含んでいるが、新しく作成された推論規則は論理演算とは異なっている。この理由は、もとの定性推論が値の大きさまで考慮したものであり、例えば $[+]$ を入力した NOT の出力は一度に $[+]$ ($[true]$) から $[0]$ ($[false]$) に変化するのではなく、 $[+] \rightarrow [s+] \rightarrow [0]$ と変化することによる。このため NOT は 2 入力回路 (本当の入力と前の時刻の出力値) で、 $[+]$ を入力すると $[s+]$ が出力されるといった推論規則が生成されていた。

Fig. 1 に示した桁上げ連鎖回路が論理回路として動作するには動作時間が必要である。人間はこのような場合、途中の変化を無視するといった一種の「近似」操作を行って対象を認識していると思われる。そこで、「結果に矛盾を生じない」という条件のもとで、推論過程を表したグラフの一部を無視する機能を付加し実験を行った。具体的には次の 2 種類の近似操作 (Fig. 2) を行うプログラムを作成し実験を行った。

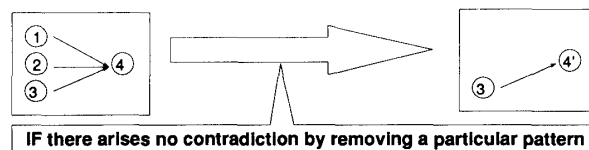
TYPE I: 推論規則に対応した色情報が特定パターンの場合に、パターン中特定部分の辺を削除。これは、特定の構成を持つ機器が果たす役割を

仮定することに相当する。

TYPE II: データの値に対応する色が特定パターンですべて同じ値ならば、その部分の辺を削除。これは、デフォルト値を仮定することに相当する。

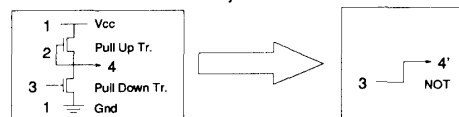
Fig. 3 に TYPE I のもののみ実行した結果を示す。ここで、

グラフの大きさ =
 点の数 + 辺の数
 + \sum 推論規則の大きさ 1
 + \sum 推論規則の大きさ 2
 - Pattern によるグラフの書換えが起こった



TYPE I : Utilize Rule Color

→ Assume a Function of Object



TYPE II : Utilize Value Color

→ Assume a Default Value

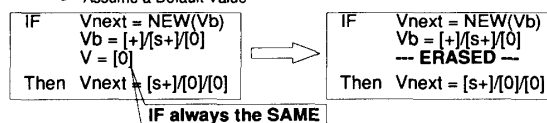
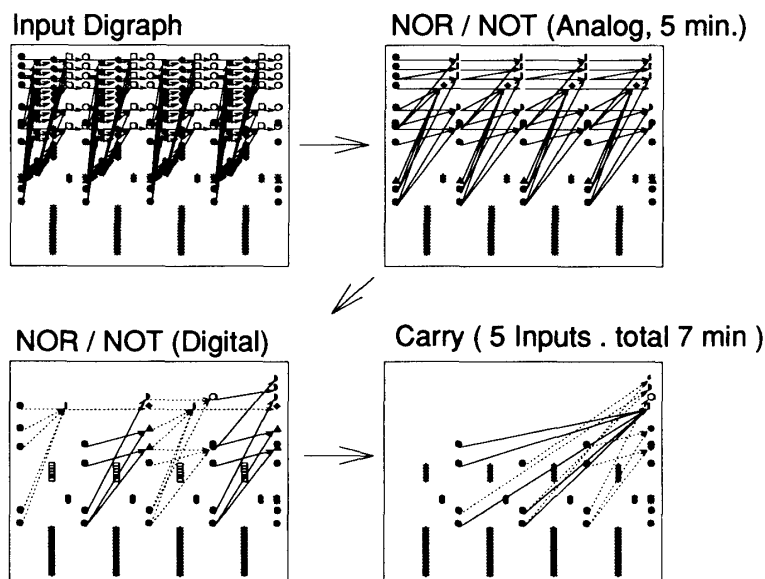


Fig. 2 Approximation.



Rules for Logical Operation (Truth Table)" found by Assuming
 a Function (unknown to CLIP) of Pullup Transistor.

Fig. 3 Importance of approximation (TYPE I).

回数

−10 * TYPE I の近似を行う Pattern による
グラフの書換えが起こった回数 (2)

パターン幅 W は 4, パターンの深さ D は 6, View 内の Pattern の最大数は 7, 同一世代内の View の最大数は 15 を使用した。50 世代のパターン抽出操作で, グラフの大きさが最低と評価されたのは 23 世代目で, 結果は近似操作を実行しなかったもの⁽¹⁾と変わらなかった (Fig. 3 右上)。これは, 初めのグラフの辺に対応している回路方程式では, まとまった機能を表すには小さすぎるため, 近似操作が起こらなかったことによる。この結果得られたグラフを再度類型パターンの抽出操作で処理した結果が Fig. 3 左下である (実線で 1 入力のパターンが NOT, 2 入力 NOR, 点線はそれ以外の固まりで NOT/NOR の学習にとってはノイズとなっている)。ここではプルアップトランジスタや電源などの回路中での機能が仮定されている。結果として得られた NOT に関する推論としては「 $[+] \rightarrow [0]$ 」および「 $[0] \rightarrow [+]$ 」の 2 種類が抽出された。これは $[+]$ を $[true]$, $[0]$ を $[false]$ と読み換えれば, NOT の論理演算規則そのものと言える。NOT に関しては桁上げ演算回路の動作中, 一方の入力 (clock 信号) がすべて $[false]$ なので, NOT と同様の推計規則が得られた。これは, 入力グラフとして与えられた環境下では, NOR と NOT に対する推論規則が同じと判断されたことを意味する。さらに同じ処理を繰り返し最終的に得られたのが Fig. 3 右下である。ここでは「5 入力の桁上げ演算」が学習結果とし

て得られた (実線で 5 入力のパターン)。

実際には「桁上げ演算」は 3 入力である。さらに TYPE II の近似操作も行った結果を Fig. 4 に示す。この結果「clock 信号は常に $[false]$ 」(前述の学習結果中では時刻を変えて 2 回参照) がデフォルト値として仮定され「3 入力の桁上げ演算」が学習結果として得られた。中間の階層として得られたグラフはかなり異なった形状を示していたが, 不要と判定された部分が省略されただけで, 基本的には TYPE I の近似操作で得られた階層構造と同じ構造であった。

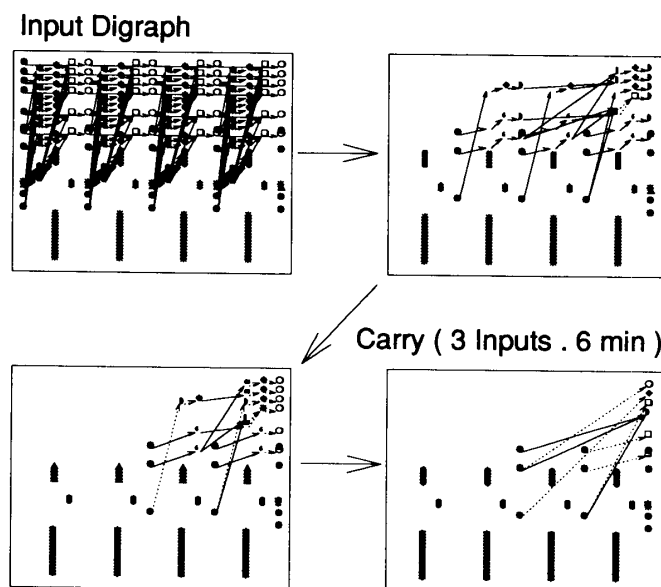
2.3 パターンマッチ能力と概念構造

文献(1)ではグラフの書換えの負荷を無視した評価式を用いた実験結果を示した。Fig. 5 に書換えの負荷を考慮した評価式を用いた実験結果を示す。グラフの大きさの評価式としては

$$\begin{aligned} \text{グラフの大きさ} = & \text{点の数} + \text{辺の数} \\ & + \sum \text{推論規則の大きさ 1} \\ & + \sum \text{推論規則の大きさ 2} \\ & + 10 * \sum \text{Pattern に含まれる点の数}^2 \\ & - \text{Pattern によるグラフの書換えが起こった回数} \end{aligned} \quad (3)$$

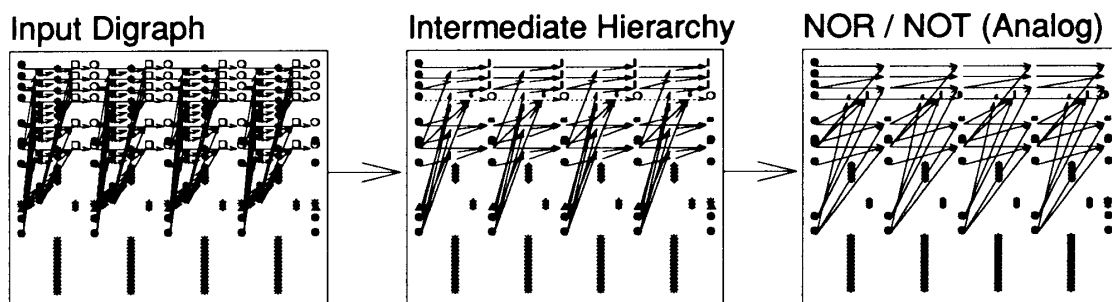
を使用した。第 5 項がマッチングコストの項である。この項が大きい Pattern ほどマッチングコストがかかることを想定している。

50 世代のパターン抽出操作で, グラフの大きさが最低と評価されたのは 15 世代目で, 一つ一つのパターン



Input Carry" found by Assuming a Default Value for the Clock Signal

Fig. 4 Importance of approximation (TYPE II).



Higher Matching Cost resulted in an Intermediate Level of Hierarchy

Fig.5 Importance of matching cost.

はマッチングコストを考慮しなかったもの⁽¹⁾より小さく、プルアップトランジスタの動作を1固まりとして認識するパターン（個々の素子レベルの知識）などが抽出された (Fig. 5 中間)。この結果得られたグラフを入力として再度パターン抽出操作をしたところ、マッチングコストを考慮しなかったものと同じグラフが得られた (Fig. 5 右端)。

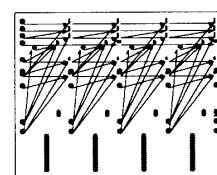
この結果は推論システムの能力特性により最適な概念構造が変化することを示唆している。すなわち、パターンマッチ能力が強力な推論システムであれば、1階層で認識できる対象が、パターンマッチ能力の弱いシステムでは複数階層で認識されたほうが全体としての負荷が少ない可能性がある。CLIP はグラフの大きさの評価式を変更することで、このような推論システムの能力特性に適した階層構造を生成できる。

2・4 「有効グラフの色」と学習結果

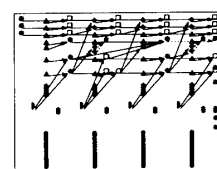
本節では、学習時に利用する情報（グラフの色）として推論規則以外のものが適用できる可能性を指摘する。Fig. 6 にグラフの色として推論規則を利用したもの (Fig. 6(a)), 全体のうち一部のみ、全体として同様のグラフが計八つ)、データ属性(データが電圧値か電流値かを示す)を利用したもの (Fig. 6(b)), 同)、およびデータ値を利用したもの (Fig. 6(c)) 全体の結果を示す。グラフの大きさの定義式など探索に使用したパラメータはすべて同じである。

すでに説明してきたようにグラフの色として推論規則を使った場合、EBL であろうところの Macro Rule が抽出できた (2・1 節および文献(1))。グラフの色としてデータ属性を使った場合、「電圧が決まるとトランジスタに流れる電流が決まる」といったパターンが抽出された。直接 EBL の Macro Rule に相当するものは

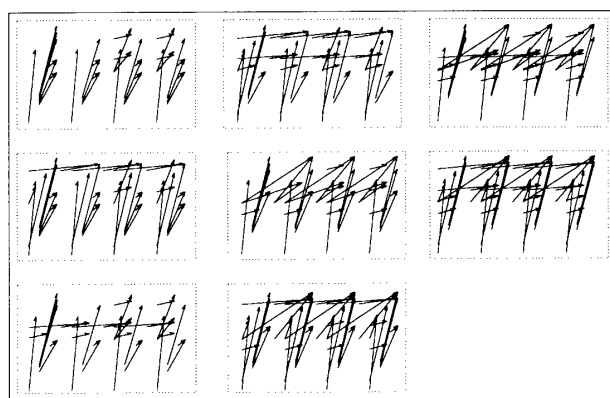
* 1 具体的にある演算をしたときの回路上での値の分布パターン。例えば、Fig. 6(b)右端の二つのグラフでは同じような計算が行われているのがわかる。



(a) Resulting digraph (subgraph) with rule.



(b) Resulting digraph (subgraph) with data attribute.

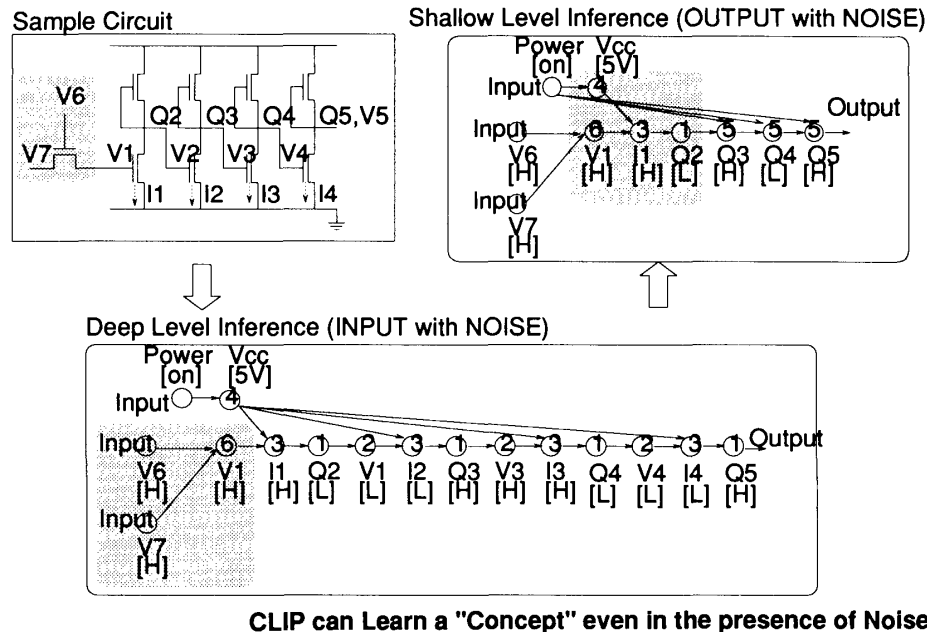


(c) Resulting digraph with value.

Fig.6 Importance of color.

抽出できなかったが、同じ一つのパターン（例えば NOR 回路にも NOT 回路にもマッチするパターン）として抽出されたものをうまく分類すれば複数の Macro Rule (グラフの色として推論規則を使った場合に得られた NOR と NOT) に分解できた。このことは、対象とした回路の場合、データ属性と関連する回路方程式の形に強い相関関係があるためと思われる。

グラフの色としてデータ値のみを使った場合、Macro Rule に相当するものは抽出できなかった。この場合でも、グラフ全体を眺めると、何らかのパターン*1 が抽出されたことがわかる。この結果からは



CLIP can Learn a "Concept" even in the presence of Noise.

Fig. 7 Effect of noise.

Macro Ruleのような推論（この場合シミュレーション）の効率化に有効な情報は得られなかった。

以上の結果はグラフの色として推論規則以外にも使用可能なものがあることを示唆している。

2・5 ノイズと学習結果

「人間は環境にノイズを含む場合にも概念を学習・生成できる。ここまで示してきた例題はいずれもその点で不十分である」という指摘はおそらく正しい。現在までに実験的に示せた例題はいずれも人間の扱っている対象よりノイズが少ない学習の容易なものである。しかし、CLIPは環境にノイズを含む場合でも理論的には学習可能である。

Fig. 7 にノイズを含んだ問題を示しこの点を考察する。Fig. 7 に示した回路は、入力が通過制御トランジスタにより与えられている。この記述では、ほかの部分で電圧・電流の関係として細かく記述されている（推論規則 1, 2, 3, 4）のに対して、通過制御トランジスタの記述だけは、途中の電流の変化などを省略して入力電圧から直接出力電圧へと大まかに記述されている（推論規則 6）。記述のレベルが揃っていないことが、概念学習にはノイズとして影響を与える。

このような場合にもノイズによる影響がほかの部分に比較し少ないうち（具体的にはグラフの大きさの評価値に大きな影響を及ぼさない場合）は、概念学習が可能である。例えば Fig. 7 の例では雑音が全くない場合に学習されるのと同じ推論規則 5 を学習している。推論規則 6 による推論のようなノイズが多い場合は少ない場合と同じ概念を記憶しても推論効率の向上には

役立たない。したがって何も学習できないなど学習能力が低下するが、「推論を簡単にする概念を見つける」という CLIP の意図から見れば問題とはならない。逆にノイズの量が極端に多い場合、類型的なノイズのパターン（例えば Fig. 7 で推論規則 6 → 3 のパターン等）を概念として学習し、結果として構成される概念構造が別のものになる場合も考えられる。

また、Fig. 3, Fig. 4 に示した例においても、最終的に回路の出力に影響を与えない場所（例えば Fig. 1 に示した桁上げ演算回路右下の NOT）での回路動作に関する推論過程は、「桁上げ演算」を学習する場合にはノイズになっており、完全にノイズを除去した実験とはなっていない。このこともノイズを含んだ環境においても CLIP が概念を学習可能であるという上記考察が正しいことを示唆している。

2・6 CLIP の特徴

以上説明してきた概念学習方法 CLIP の特徴をまとめると以下の 10 点となる。

1. 推論過程の分析結果から推論を効率的に行うという点で有効な概念を生成・学習する概念学習の方法である。
2. あらかじめ概念を形成するための特別の知識を用意しておかなくても「推論を簡単にする」単語として概念が生成できる。
3. 生成した概念を用いた新しい推論規則により、推論を高速化できる。
4. 例題からの学習方法ではなく、学習・推論システムが置かれた環境からの学習方法であり、例題を

作成する「教師」は不要である⁽¹⁾。

5. 色つき有向グラフ上で定義された学習方法であり、「近似」などの操作も色つき有向グラフの操作として定義されており、特定の専門領域に依存しない⁽¹⁾。
6. グラフの色として推論規則を用いた場合、Goal Concept と Operational Criteria を自動的に調整することで Utility 問題に対応し、Unification の代わりに辺の対応関係を利用して高速化した EBL システムとみなせる⁽¹⁾。
7. 「近似」の考えを導入し推論過程を表したグラフの一部を無視することで、論理演算規則など上位レベルの概念を合成できる。
8. 推論システムの能力特性(例えば、パターンマッチング能力)に合わせた、適切な概念構造を形成できる。
9. 学習時に利用する情報(グラフの色)として推論規則以外にもデータ属性などが利用できる。
10. 環境にノイズを含む場合でも学習可能である。

3. 関連研究との比較

3・1 帰納推論との比較

従来概念学習といえば、対象がある概念を満足するか否かの分類木を正例・負例より学習する帰納推論の研究(例えば INDUCE/ID 3/Version Space⁽⁴⁾)が多く、学習した概念の利用まで含めて考えた研究は少なかった。

CLIP では「概念」を「利用することにより推論が簡単になる言葉」であると捉え、推論過程からの概念学習を考えている。学習結果の「概念」を利用した推論の効率化まで考えた点がほかの研究と大きく異なっている。CLIP によれば、例えば電気回路を対象とした電圧・電流の変化に関する定性推論の過程を分析し、推論過程を簡単にする単語として、NOR/NOT に相当する単語を自動的に生成・学習できる。あらかじめ論理演算に相当する知識を入力しておかなくても「推論を簡単にする」単語として類似の概念が生成され、合成した推論規則により推論を効率化できる点が特徴である。

また、近年帰納推論と強い関連を持つ研究として COBWEB⁽⁵⁾ など Conceptual Clustering の研究が盛んになりつつある。COBWEB は未知データに対して

Table 1 The differences between COBWEB and CLIP.

	COBWEB	CLIP
Input	<ul style="list-style-type: none"> • A Set of Object Descriptions 	<ul style="list-style-type: none"> • A Set of Inference Patterns • Domain Theory • Characteristics of Inference Engine
Output	<ul style="list-style-type: none"> • Useful Sets of Objects • Their Conceptual Descriptions 	<ul style="list-style-type: none"> • Useful Sets of Chunks (Composite Objects) from Inference Patterns • Conceptual Descriptions to all Chunks
Note	<ul style="list-style-type: none"> • Maximize Category Utility : To Predict Unknown Properties 	<ul style="list-style-type: none"> • Minimize Graph Size : To Improve Efficiency

発見した概念の暗黙値を用いた推察を可能とする (Table 1) 点で、学習した概念の利用まで含めて考えた研究であるが、CLIP のような推論の効率化の機能は持っていない。

「概念」を「利用することにより推論が簡単になる言葉」であると捉えることにより、学習結果の利用が可能になった点が CLIP の最大の特長である。

3・2 AM との比較

いわゆる帰納推論ではないが、自動的に概念形成を行った研究として AM⁽⁴⁾ がある。AM は概念形成のための特別の Heuristics を複数用意しておくことで数学の公理から出発して、初めの公理には含まれていなかった概念を生成する能力を持っている。

CLIP ではこのような特別の Heuristics は不要である。探索を制御するものとして、入力として与えるグラフおよびグラフの大きさの計算式があるが、前者は推論システムの置かれた環境を定義するもの、後者は推論システムの能力特性を定義するものであり、Heuristics ではない。見方を変えると CLIP は「小さなグラフを与えるものが概念である」または「与えられた環境の中で推論システムの効率を上げるものが概念である」*2 という特別な Heuristics を一つだけ使った学習システムともみなせる。

種々雑多な領域に応じた Heuristics を用意することは困難な作業である。多数の Heuristics に頼る手法に比べ他分野への適用は容易であると考えている。

3・3 EBL との比較

CLIP はグラフの色として推論規則を用いた場合、Goal Concept と Operational Criteria を自動的に調整することで Utility 問題に対応し、Unification の代わりに辺の対応関係を利用して高速化した EBL⁽²⁾⁽³⁾ システムとみなせる⁽¹⁾。Utility 問題に関連した EBL

* 2 グラフの大きさの定義より、「小さなグラフを与えるもの」と「与えられた環境の中で推論システムの効率を上げるもの」とは同じ意味である。

の研究としては文献(6) - (10)など重要なものが多い。

Lebowitz の UNIMEM⁽⁶⁾ は出現頻度の低いものから出現頻度の高いものの説明を作成し、作成可能な場合一般化して記憶する仕組みを持っている。例えば議会の投票記録から、「MX ミサイルの削減に反対する議員は防衛費の削減に反対する傾向が強く、軍関係の支出の多い選挙区であることが多い」ことを見つけてくる。これは一種の概念を発見したものとみなせるが、CLIP のように見つけた概念が効率的な推論を実現するよう、種々のパターンを調べ、適切なものを選ぶ機能は持っていない。

Cohen の (Approximating) Abductive EBL⁽⁷⁾ は candidate pool 内に登録しておいた Macro Rule の中から訓練例を最も効率良く解く Macro Rule の組合せを求める手法である。これは candidate pool が考えられるすべての Macro Rule を含めば CLIP と同じ手法とみなせる。ただし、一般に考えられるすべての Macro Rule を candidate pool に含めることは不可能である。例えば桁上げ演算回路の場合、入力グラフの辺の数は 2000 を超えていた。したがって考えられる部分グラフへの分割方法は、 $2^{\text{入力グラフの辺の数}} > 2^{2000}$ である。これらの中から同じものを削除する等の処理を経て力まかせに candidate pool を作ることは不可能である。CLIP はこの過程を文献(1)で説明した並列探索手法で自動化したものとみなすこともできる。

Minton の PRODIGY⁽⁸⁾ も個々の Macro Rule の Utility を評価し、効果的なもののみ記憶している。山田の Perfect Causality⁽⁹⁾ は「どのような形をした部分グラフが Macro Rule として Utility が高いか」に関する Heuristics である。PRODIGY のように個々の Macro Rule の Utility を個別に評価しても全体の評価値が良くなる保証はない。また Perfect Causality は方程式問題の解決に有効な Heuristics であることが実験的に確かめられているが、適用範囲は明らかになっていない。

山村らは複数の説明木の根から共通構造に着目して Operational Criteria を調整し、Utility の高い Macro Rule を取り出す手法を提案している⁽¹⁰⁾。CLIP では木構造の途中に含まれる共通部分の構造も取り出している。

以上説明してきた EBL、特に URility 問題の研究は

* 3 TYPE I の近似操作は、Pattern Modification の段階で見つけたパターンの前の部分（推論の前のほうの段階で利用されたルールに対応する Pattern の一部分）を無視する（ように補助的なマークのついた）Pattern と無視しない普通の Pattern を作成することで実現した。Type II の近似操作は Pattern によるグラフの書換え後、推論規則表を作る段階で実行した。

Table 2 The differences between EBL and CLIP.

	EBL	CLIP
Input	<ul style="list-style-type: none"> • Goal Concept • Training Example • Domain Theory • Operability Criterion 	<ul style="list-style-type: none"> • Plural Training Examples • Domain Theory • Characteristics of the Inference Engine
Output	<ul style="list-style-type: none"> • A Macro Rule 	<ul style="list-style-type: none"> • A Set of Macro Rules (Operability Criterion Implicitly Embedded) • Interpretation Rule : (New Domain Theory for the Abstract Level Inference) • Goal Concept

いずれも概念学習を意図したものではない。そのため、学習した概念を用いた推論を行うことはできない。一方 CLIP では変換結果の概念を利用して推論を行うための推論規則 (Fig. 1 Interpretation Rule) まで出力するので、新しい概念を使った推論を行うことができる。例えば CLIP では、電圧や電流の関係を記述した回路方程式を Fig. 1 に例を示した Macro Rule を用いて論理レベルの記述に書き換え、書き換えた記述を使って論理レベルの推論（例えば「not の入力が [[false] だったので出力は [true]]」を行うことができる。EBL に限らず、チャンキングを利用した機械学習の研究でこのような機能を実現した例は見当たらない。例えば文献(2)の例では、“handle”や“light”から構成された記述を、“cup”という上位レベルの概念に変換することまではできるが、レベルが変わった世界(“cup”で記述されなおした世界)での推論（例えば“cup”を使って水が飲める）までは扱っていない。出力に Interpretation Rule を含むこと (Table 2) が CLIP の大きな特徴である。

3・4 並列探索手法

CLIP の類型パターンの抽出操作は一種の Genetic Algorithm⁽¹¹⁾ とみなせる⁽¹⁾。CLIP の類型パターンの抽出操作の特徴は Mutation に相当する Pattern Modification が強力な点である。Fig. 8 に Fig. 4 の実験時に世代ごとに得られたグラフの大きさと、世代ごとの View が Pattern Modification で作成されたものか、Pattern Combination で作成されたものかの比を示す。Type I の近似操作は Pattern Modification の一種として実現されている。Fig. 8 下部で白塗りの部分だけが Pattern Combination による操作で、後は Type I の近似操作も含めて Pattern Modification により作られた View の比率である*3。また、Fig. 8 上部

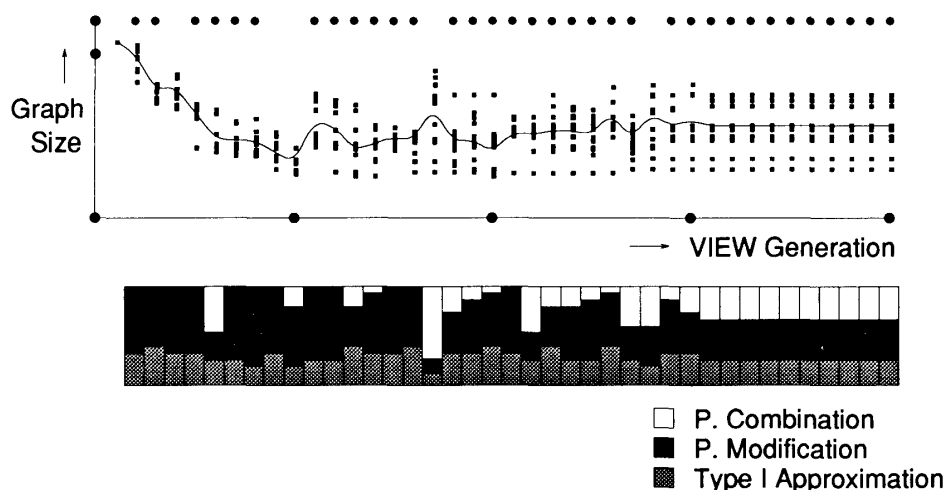


Fig.8 Log of search (with approximation).

で上端の黒丸は近似操作を行った結果矛盾が生じ、作成した View の中に捨てられたものがあつたことを意味している。

Genetic Algorithm の研究では処理効率を上げることが一つの研究課題となっているが、CLIP の実験結果は Mutation (CLIP では Pattern Modification) を工夫することにより処理効率を上げることが可能な事例の存在を示している。

また、現在実験に利用しているプログラムは確率的な処理を行っておらず、その意味では INDUCE⁽⁴⁾ や音声理解システム HARPY⁽¹²⁾ に利用された beam search に近い自然淘汰を利用した並列探索システムになっている。

3・5 階層的知識ベース

エキスパートシステムの高機能化に伴い知識ベースが大規模になりその管理技術の重要性が認識されつつある。そのような研究の一環として著者らは、大規模・複雑な対象に関する知識を上位レベルの知識（機器の仕様に相当）と下位レベルの知識（その実現に相当）に分類し階層的な知識ベースにまとめる研究を行っている⁽¹³⁾⁽¹⁴⁾。

上位レベルと下位レベルの知識をつなぐためには、「近似」が重要であることはこれらの研究の過程で得られた知見であり、CLIP は「近似」まで含めた処理全体を自動化しようという発想から考案された手法である。自動化することにより階層的な知識ベースを作成するのに必要な入力量を削減することが可能になった。

これらの技術は対象に関する知識の収集・整理技術であり、将来的にはタスク知識の整理・収集技術と統合されるべきものである (Fig. 9)。

Functional Description

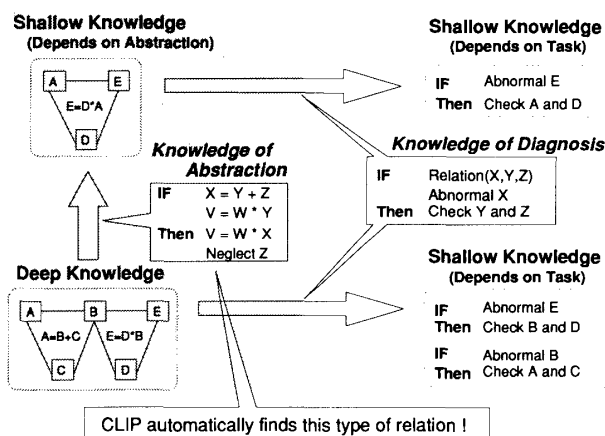


Fig.9 Image of hierarchical knowledge base.

4. 今後の課題

CLIP に関して今後に残された課題を以下にまとめる。

(1) 実装上の課題

- ・現在実験に使用しているプログラムは Type I の近似操作をした後の Pattern を Pattern Modification の対象にしない。これは現在の実装方法では近似操作を行った前後のグラフの接続情報が不明になるためである。このため多重階層の概念構造を作成した際に不必要な階層が作成される場合があった。この実装方法は改善可能である。
- ・現在実験に使用しているプログラムは 1 出力の概念しか取り扱えない。桁上げ演算回路の場合「桁上り」と「その桁の値」の 2 出力回路とみなせるが、現在は 1 出力の概念の組合せとしか取り扱えない。多出力概念の扱いは今後の課題である。

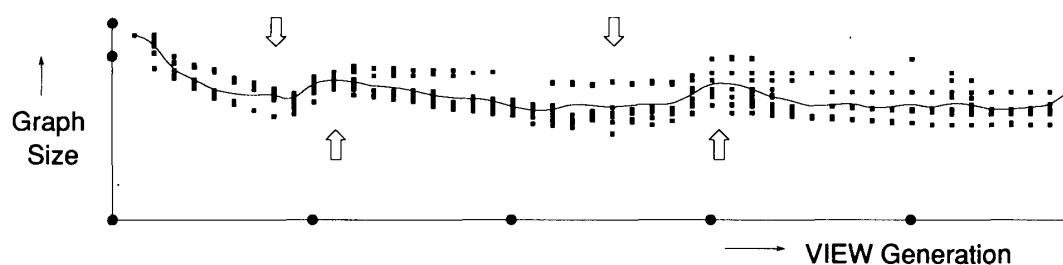


Fig. 10 Log of search (without approximation).

(2) 階層型知識ベース構築と関係した課題

- 現在の CLIP は 2 階層の知識ベースであれば、推論システムの能力特性に合わせて適切な構造を探索する。この探索を繰り返すことで多重階層の知識ベースも構築可能である。しかし、この場合全体を何階層に分けるかまで含めて最適化する機能はない。今後の重要な研究課題である。
- グラフを単純化した場合にもとのグラフに含まれていたデータが上位概念の説明には不要なものとして書換え後のグラフに含まれなくなる場合がある。このとき現在の CLIP はデータごとの重要度を全く考慮していない。実際にはデータの観測可能性、制御可能性などによりデータごとの重要度が定まり、それによって残すべきデータが影響されると思われる。これらの考慮も今後の課題である。
- 現在までの実験では推論規則はシミュレーション用のもののみ使用した。診断など他タスクのための推論規則を用意し、タスク知識の整理・収集技術と統合する必要がある。
- CLIP ではグラフの大きさの評価式として与えるパラメータを調整することで、利用する推論システムに合わせて推論システムの処理効率を最適化することができる。しかし、推論システムによっては作成した概念構造が人間の概念構造と一致しない場合が予想される。例えば 2・3 節で考察したパターンマッチ能力と生成した概念構造の関係は、このような場合の一例に思える。計算機と人間の協調システムを考えるうえでは両者が一致することが望ましい。作成した概念構造を利用者が取捨選択する機構や両者が一致しやすいパラメータなどの検討を行う必要がある。

(3) その他の課題

- 現在近似操作を実現するために値に対応した色を利用している。近似してよいか否かの判断として、すべての場合について矛盾を生じないか否かの確認をしているが、この方法が使えるのは値が離散値でとり得る値の数が小さい場合のみである。人

間であれば証明用の知識(例えば数式の変形知識)を別に持っていて、それによる証明を行っている場合も多いと思われる。このような証明過程の実現は今後の課題である。

- 現在の CLIP の枠組みは逐次的に学習結果を改良していく仕組みを持たない。概念を小修正して環境の変化に追従させる機能は今後の検討課題である。
- 現在実験に利用しているプログラムは確率的な処理を行っていない(3・4 節参照)。このため、Genetic Algorithm が持っていると期待される局所最適解からの脱出能力が弱い。これ以外にも Pattern Modification では長いパターンへの変形しにくいといった欠点もある。例えば、Fig. 10 に Fig. 1 の結果を得た実験時に、View で書き換えた後のグラフの平均的大きさを計測した結果を示す。Fig. 10 の結果は、左端の下向き矢印のところで短いパターンによる局所的な解を見つけた後、パターンが長くなり一端結果が悪くなった(図上向き矢印)後、中程度の長さのパターンを右側下向きの矢印の部分で見つけたことを示している。短いパターンのときにさらに良い解があったとしても後ろの世代で見つかる可能性は少ない(View の記憶容量があふれ長いパターンがすべて捨てられ、再度短いパターンが探される可能性が 0 ではない)。探索手法としては改良の余地が多いと思われる。

5. む す び

推論過程の分析結果から推論を効率的に行うという点で有効な概念を生成・学習する概念学習の方法 (CLIP: Concept Learning from Inference Pattern) に関して考察を加え、以下の結論を得た。

- CLIP は「環境に含まれている固まりを見つける」という意味での「概念生成能力」を持つ。
- 論理回路の電氣的な振舞いから論理演算規則を合

成するには「近似」の考えを導入し、推論過程を表したグラフの一部を無視することが必要である。

- ・推論システムの能力特性（例えばパターンマッチング能力）により適切な概念構造が変化する。
- ・学習時に利用する情報（グラフの色）として推論規則以外にもデータ属性などが利用できる。
- ・CLIP は環境にノイズを含む場合でも学習可能で

ある。

謝 辞

竹内勝氏（日立製作所基礎研究所）には Genetic Algorithm と CLIP の関係について御討論いただきました。杉田智弘氏（同）には実験プログラムの開発に御協力いただきました。ここに謝意を表します。

◇ 参 考 文 献 ◇

- (1) 吉田, 元田: 推論過程からの概念学習(1)類型的推論過程の抽出, 人工知能学会誌, Vol. 7, No. 4, pp. 675-685 (1992).
- (2) Mitchell, T. M., Keller, R. M. and Kedar-Cabelli, S. T.: Explanation-Based Generalization: A Unifying View, *Machine Learning*, pp. 47-80 (1986).
- (3) DeJong, G. and Mooney, R.: Explanation-Based Learning: An Alternative View, *Machine Learning*, pp. 145-176 (1986).
- (4) Cohen, P. R. and Feigenbaum, E. A. ed.: *The handbook of artificial intelligence*, chap. 12, William Kaufmann, Inc. (1982).
- (5) Fisher, D. H.: Knowledge acquisition via incremental conceptual clustering, *Machine Learning*, pp. 139-172 (1987).
- (6) Lebowitz, M.: Integrated Learning: Controlling Explanation, *Cognitive Science*, Vol. 10, pp. 219-240 (1986).
- (7) Flann, N. S.: Applying Abstraction and Simplification to Learn in Intractable Domains, *ML-90*, pp. 277-285 (1990).
- (8) Minton, S.: Quantitative Results Concerning the Utility of Explanation-Based Learning, *AAAI-88*, pp. 564-569 (1988).
- (9) Yamada, S. and Tsuji, S.: Selective Learning of Macro-operators with Perfect Causality, *IJCAI-89*, pp. 603-608 (1989).
- (10) 山村, 小林: EBL の複数例題下への拡張, 人工知能学会誌, Vol. 4, No. 4, pp. 389-397 (1989).
- (11) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company (1989).
- (12) Barr, A. and Feigenbaum, E. A. ed.: *The handbook of artificial intelligence*, chap. 5, William Kaufmann, Inc. (1981).
- (13) Yoshida, K. and Motoda, H.: Hierarchical Knowledge Representation based on Approximation, *JKA W90*, pp. 345-360 (1990).
- (14) 吉田, 元田: 階層的定性推論のための浅い知識の合成法, 人工知能学会誌, Vol. 4, No. 4, pp. 447-455 (1989).

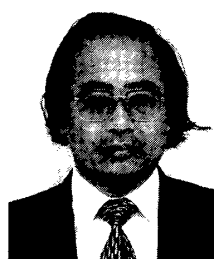
[担当編集委員: 佐藤泰介, 査読者: 半田剣一]

著 者 紹 介



吉田 健一（正会員）

1980 年東京工業大学理学部情報科学科卒業。同年、(株)日立製作所に入社。同社エネルギー研究所にてプラントの異常診断などの研究に従事。1986 年より、基礎研究所にて知識表現、機械学習、定性推論などの研究に従事。1984 年日本原子力学会論文賞、1990 年電気学会論文賞、1991 年人工知能学会全国大会優秀論文賞受賞。情報処理学会、AAAI、ACM 各会員。



元田 浩（正会員）

1965 年東京大学工学部原子力工学科卒業。1967 年同大学院原子力工学専攻修士課程修了。同年、日立製作所に入社。同社中央研究所、原子力研究所、エネルギー研究所を経て、現在、基礎研究所主管研究員。原子力システムの設計、運用、制御に関する研究、診断型エキスパートシステムの研究を経て、現在は人工知能の基礎研究、特に機械学習、知識獲得、知識コンパイル、定性推論などの研究に従事。工学博士。元日本ソフトウェア科学会理事、現人工知能学会理事、Knowledge Acquisition (Academic Press) 編集委員、IEEE Expert 編集委員、コンピュータ科学編集委員。1970 年日本原子力学会奨励賞、1977、1984 年日本原子力学会論文賞、1991 年人工知能学会論文賞受賞。情報処理学会、日本ソフトウェア科学会、日本認知科学会、日本原子力学会、AAAI、IEEE Computer Society 各会員。