帰納的推論の記述に有用な知識表現の一提案†

堀 浩一†† 斉藤忠夫††† 猪瀬 博††

本論文は帰納的推論の記述等を目的とした H-net (Hierarchical network) とよぶ新しい知識表現法を提案している．H-net は，1階述語論理に理論的基礎をおいているが，知識の表現に必要な能力を具備するように考案されており，概念を，外延と内包を区別して表現することを特徴としている．本論文では，H-net の表現と定義を述べ，その解釈を説明する．これによって H-net はフレームと同様の表現能力をもつことが示される．H-net を解釈し操作するための言語として HNRL (H-net Representation Language) を開発実装した．本論文ではそれによる実験例についても述べている．

1. まえがき

自然言語処理システム，問題解決システムなど，知的な処理を行うシステムにおいては，知識を表現し，蓄積し，操作する機能が不可欠である．さらに人工知能の研究においては，知識の獲得に関する方法論の探求も重要な課題となっている．この問題の解決のため，従来，フレームやセマンティック・ネットワークなどのいくつかの知識表現法と，それらを操作するためのシステムが提案されている[1]．しかし，実際に，知識を利用するシステムを作成しようとするとき，従来の知識表現に対して，いくつかの不満を指摘することができる．

第1の不満は，帰納的推論の記述が容易でないことである．与えられた公理から事実の真偽を推論することを演繹的推論，与えられた事実の集まりから公理を推論することを帰納的推論とよぶとすれば，従来の知識表現はすべて演繹的推論の記述に適した表現であるということができる．知識システムの取り扱う知識に関わる原理あるいは性質がすべて把握されているときには，従来の知識表現とその操作システムを用いて，全体のシステムを構築することが可能である．しかし，自然言語理解システムなどの作成においては，あらかじめ，蓄えるべき知識の性質をすべて把握することができないのが普通である．換言すれば，個々の事実を蓄えておき，逐次，知識を整理し，原理を抽出してゆく機能が，知識システムには望まれるのである．この問題は，知識獲得過程の研究，知識システム作成のための環境の研究，帰納的推論の理論とアルゴリズムの研究[2]，あるいは類推の理論の研究[3]などと関わる問題であるが，さまざまな視点から論じ，改善することが必要であろう．本論文では，帰納的推論の問題を，知識表現として，記述することが容易か否かという観点からとらえて検討を行う．

従来の知識表現に対する第2の不満は，知識のセマンティクスを記述しようとするとき，従来の知識表現においてはその定義と記述が必ずしも形式的に明確でないことである．フレームにおいては手続き付加という形で取り扱われているし，セマンティック・ネットワークにおいてはセマンティクスの大部分は，インタプリタの解釈の仕方によって定まるといってよい．知識表現において，セマンティクスの定義と記述はそれ自体大きな問題であるから，何らかの理論的裏づけをもつ知識表現法が要求されるのである．

本論文では，上記の不満を解消する目的で H-net (Hierarchical network) と称する知識表現を提案する．知識表現にあたって，帰納的推論の記述を容易にするためには，事実と原理がそれぞれ明確に記述でき，それらの間の論理的な関係が明確に定められていなければならない．そのような表現方法を考えるとき，現在のところ最も理論的な裏づけがはっきりしているのは，1階述語論理の論理体系であろう．H-net は1階述語論理におけるモデルの表現を基本とした表現である．1階述語論理を表現の理論的基礎とすることにより，第2のセマンティクスの記述の形式的明確さの問題も同時に解決することができる．

1階述語論理に基づいた言語としては Prolog が注目を集めている．前記の不満の解消を Prolog に求めることも可能ではあろう．しかし，知識の表現のためには，Prolog だけでは記述能力に不足がある．また，一方ではフレームの記述に述語論理に基づく記述を部

† An Approach to Knowledge Representation for Describing Inductive Inference by KOICHI HORI, TADAO SAITO and HIROSHI INOSE (Faculty of Engineering, University of Tokyo).

†† 東京大学工学部電子工学科
††† 東京大学工学部電気工学科

分的に導入しようとする試みもある[4]が，手続き付加を論理式におきかえただけでは，前記の不満を解消できそうにない．H-net は Prolog や Lisp などのプログラム言語と，フレームやセマンティック・ネットワークなどの知識表現との中間に位置する表現である．すなわち H-net を用いて，フレームやセマンティック・ネットワークなどの知識表現を組み立てることができる．また，Prolog が1階述語論理における言語の記述であるのに対して，H-net は基本的にはモデルの記述であり，階層構造をなした世界の概念など，Prolog にはない記述能力をもつ．

H-net は一つの概念を，概念の外延と内包を与えることにより記述する．また，知識の最小構成要素としての primitive を陽に記述する．2章で H-net の表現の定義を与える．3章で H-net の解釈について，基本的解釈とフレームを組み立てるための解釈について述べる．4章で H-net を操作するための言語 HNRL (H-net Representation Language) について述べ，5章で H-net の応用について述べる．

2. 知識表現 H-net の表現

1階述語論理においては，1階述語論理の言語で表現された一群の文からなる理論を，真ならしめる解釈をモデルとよぶ．モデルは順序組 $\langle A, R_0, R_1, \cdots, R_m, G_0, G_1, \cdots, G_n, x_0, x_1, \cdots, x_q \rangle$ で表すことができる．Aは実体のドメインを，$R_0, \cdots, R_m$ は実体間に成立する関係を，$G_0, \cdots, G_n$ は関数を，$x_0, \cdots, x_q$ は実体を表す[5),6)]．H-net は，このモデルを出発点として，帰納的推論の記述の容易さも念頭において，表現の拡張を行い構成された知識表現である．

H-net はノードとリンクと世界から構成される．これを**表1**に示す．ノードには，primitive node, concept node, set node, structure node, lambda expression node の5種類がある．リンクには，extension link, intension link, element link, part link の4種類がある．

表 1 H-net の基本的構成要素
Table 1 Basic organizing elements of H-net.

ノード	primitive node
	concept node
	set node
	structure node
	lambda expression node
リンク	extension link
	intension link
	element link
	part link
世界	

まず，階層構成をなした世界を定める．一つの世界は，1階述語論理のモデルにおけるドメインに相当する．世界とは，自然数の世界，ある人の信じる世界など，表現された知識の適用される範囲である．下位の世界からは上位の世界が見えるが，上位の世界からは下位の世界は見えない．すべてのノードとリンクは，どこかの世界の中で定義される．

ある世界の中で，存在が直観される実体を primitive node で表す．primitive node は，1階述語論理のモデルにおける実体に相当する．たとえば，北の湖という力士を一つの実体として認識した場合，それを primitive node とする．primitive node を陽に記述する理由は，知識を表現するとき，何を知識の最小構成要素として認識しているかを明らかにするためである．

concept node は概念に名前をつけるためのノードである．concept node だけが，外部から見える名前をもっている．concept node とそれが指示する node を extension link で結ぶことにより，指示された概念に，名前をつけることができる．既存の node と extension link で結ぶことにより新たに作られた concept node も，H-net では実体と考える．1階述語論理のモデルと対応させると，primitive node だけが実体であるのに対して，extension link で結ぶことにより新たに作られたノードをも実体と考えることを，モデルの外延的拡張とよぶ．たとえば，北の湖という特定の力士を一つの実体として表現するときは，primitive node を新たに一つつくり，北の湖という名前に対応する concept node と extension link で結ぶことになる．この場合北の湖という concept node も H-net の世界上での実体となる．それが extension link で指示するノードが primitive node であることは，その実体がそれ以上分解できない実体であることを示している．このとき，primitive node は H-net で表現された世界上では，北の湖という力士を認識したときの，画像や音声などの情報とおきかわるものと考えることもできる．

set node は集合を表すためのノードである．set node とおのおのの要素のノードとを element link で結ぶことにより集合を表す．set node に対応する概念は，1階述語論理のモデルには存在しない．

structure node は構造あるいは関係を表すための

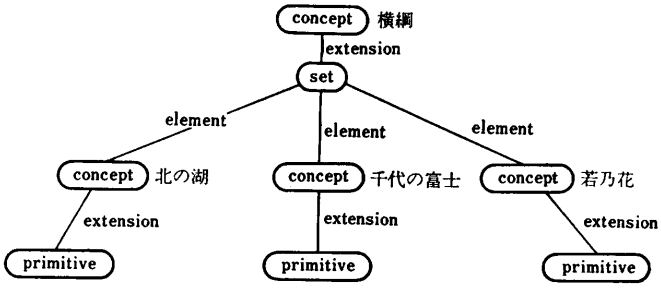

図 1 外延の表現の例
Fig. 1 An example of representation of extension.

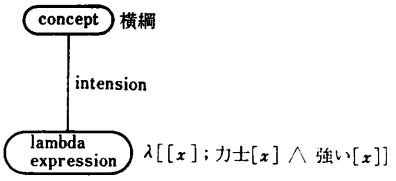

図 2 内包の表現の例
Fig. 2 An example of representation of intension.

ノードである．structure node と，構造あるいは関係の構成要素のノードとを part link で結ぶことにより構造あるいは関係を表す．structure node は 1 階述語論理のモデルにおける関係に相当する．

すでに明らかなように，extension link は概念の外延を指示するためのリンクである．外延とはある概念の実現値あるいは指示物のことである．たとえば，横綱という概念は**図 1** のように表現できる．

以上の primitive node, concept node, set node, structure node, extension link, element link, part link を用いることにより，H-net は現実世界で成立している事実のモデルを表現することができる．帰納的推論の記述を容易にするためには，事実の記述と原理の記述を明確に区別し，事実と原理の関係を明白にしなければならない．H-net では概念の内包を記述するという形で原理を表現する．内包とはある概念を定めるための属性である．外延と内包の区別の必要性は従来から指摘されているとおり[7] であり，内包的論理学のモデルと対応づけることもできる．

lambda expression node はラムダ表記された関数を表現するためのノードである．その定義域は H-net の世界上のノード，値域は真偽値とする．concept node と lambda expression node を intension link で結ぶことにより概念の内包を表現する．lambda expression node は，concept node が名前をもつのと同様に，ラムダ表記された式をもつ．ラムダ表記された式とは，H-net においては，$\lambda[[x_0 \cdots x_n]; c_0[x_i \cdots x_j] \wedge c_1[x_k \cdots x_l] \wedge \cdots \wedge c_n[x_m \cdots x_n]]$ の形をした式である．ただし，$c_0, c_1, \cdots, c_n$ はすでに定義された concept node の名前で，$c_0[x_i \cdots x_j]$ は $x_i \cdots x_j$ からなる structure node (x_i 1 個だけのときはそのノード) が c_0 というノードの外延 (またはその要素) でありうるときは真，そうでないときは偽となる述語であり，$\wedge$ は論理積を表す．H-net において概念の内包とは，あるノードがその概念の外延 (またはその要素) でありうるかどうかの判断を与える述語であり，上記のラムダ表記された式をノードに作用させることにより，この判断がなされる．

lambda expression node をつくり，concept node と intension link で結ぶことにより新しい concept node をつくることができる．これをモデルの内包的拡張とよぶ．ラムダ表記された式を表現に含むことは，モデルと言語の部分的な融合を許すことになるが，人間の知識を考えるとこれは自然な要求として容認できると考える．横綱の内包を**図 2** に示す．ただし，力士，強いなどの concept node はすでに定義されているものとする．

H-net には，1 階述語論理のモデルにおける関数に直接対応するものはない．関数の値を，引数の最後につけ加えて，関係として表すことにする．

以上で事実と原理の表現方法と，事実と原理の間の関係が定まった．

3. 知識表現 H-net の解釈

知識表現 H-net の解釈とは，H-net を用いて表現された知識が何を意味しているかという問題である．

H-net の解釈はいくつかの視点から考えることができる．H-net を 1 階述語論理のモデルとしてとらえれば，H-net は 1 階述語論理による表現に対する解釈を与えるものであると解釈できる．また，H-net を知識表現としてとらえると，H-net によって表現されたある種の知識はフレームを表現していると解釈することができる．さらに，知識表現として重要な，default, ISA hierarchy, inheritance 等の概念を H-net が自然に表現しているとの解釈も可能である．

まず，基本的な解釈として，H-net を 1 階述語論理のモデルとして考える．H-net 表現と，1 階述語論理による表現との間のやりとりをする関数として，deduce と induce という二つの関数を定義する．

deduce という関数は，H-net の表現と 1 階述語論理の表現を引数とし，1 階述語論理の表現に対する H-net 上での解釈を値として返す演繹的推論のための関数である．1 階述語論理のアトムに対しては，対応する H-net 上の実体を返す．1 階述語論理の $P(a)$ の形をした基本論理式に対しては，a に対応する実体が，P に対応するノードの外延でありうるときは真を，そうでないときは偽を返す．あるノードが別のノードの外延でありうるかどうかの判断は，まず，実際に二つのノードが extension link で結ばれているかどうかを調べ，結ばれていないときは intension link をたどり内包を評価して判断を行う．変数を含む式に対しては，変数を適当な実体とおきかえることにより真となりうるかどうかの判断を行う．

induce という関数は，H-net の表現と 1 階述語論理の表現を引数とし，H-net の新しい表現を値として返す，帰納的推論のための関数である．1 階述語論理の言語上の，一連の $P(a)$, $P(b)$, $P(c)$, …が真であると宣言されたとき，induce という関数は，$a, b, c, \cdots$ のそれぞれに対応するノードがなにゆえ，P に対応するノードの外延でありうるのかという，P に対応するノードの内包を推論し，出力する．induce はさらに，try-apply と match という二つの関数からなる．try-apply は，すでに H-net の世界上に定義された知識を，$a, b, c, \cdots$ に対応するノードに適用して，それぞれのノードがどのような内包をもっているかを探るための関数である．match はそれらの内包どうしのマッチングをとって，P に対応するノードの内包を推測するための関数である．

deduce, induce ともに，H-net 表現の解釈を行う H-net インタプリタ上に組み込まれなければならない．それらのアルゴリズムはインタプリタの作成者にゆだねられるが，文献 2), 5) などを参考にできる．実装の一例は 4 章に示す．

次に，応用的な解釈として，H-net をフレームとして解釈する方法を述べる．フレームと同様の，フレームとスロットからなる知識の構造を，H-net における 3 種類の表現に見いだすことができる．一つは structure node による表現，一つは内包による表現，他の一つは世界による表現である．逆に言えば，フレーム表現では，すべてフレームという形に縮退していた表現が，H-net 上では縮退が解けて，3 種類の表現に区別されていると考えることもできる．

まず，structure node について考える．structure node は，それ自体は現実世界には実体として存在しえないが，いくつかの構成要素が集まってはじめて存在を認識されるような構造としての実体を表していると解釈できる．たとえば，**図 3** に示すような日付の概念の H-net による表現は，**図 4** に示すようなフレーム表現と等価であると解釈できる．

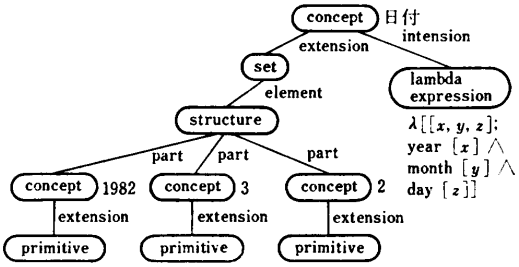

図 3　日付の概念の structure node を用いた表現

Fig. 3　Representation of the concept of date by the structure node.

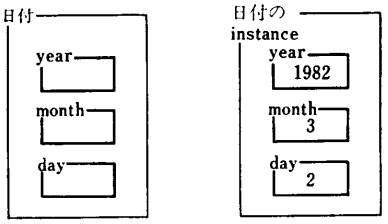

図 4　日付の概念のフレーム表現

Fig. 4　Representation of the concept of date by the frame.

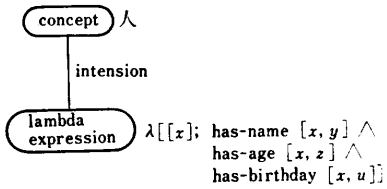

図 5　人の概念の内包による表現

Fig. 5　Representation of the concept of person by the intension.

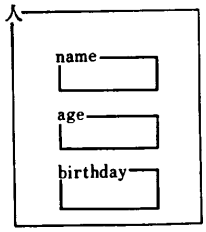

図 6　人の概念のフレーム表現

Fig. 6　Representation of the concept of person by the frame.

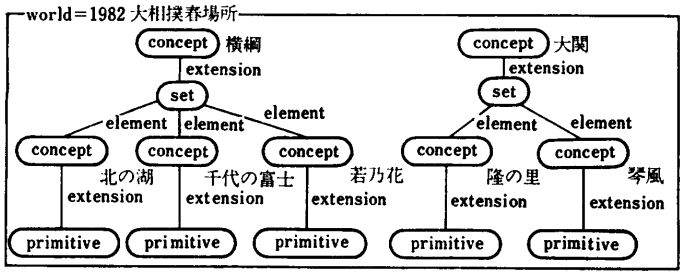

図 7 1982大相撲春場所の概念の世界による表現
Fig. 7 Representation of the concept of 1982-Sumo-wrestling-spring-tournament by the world.

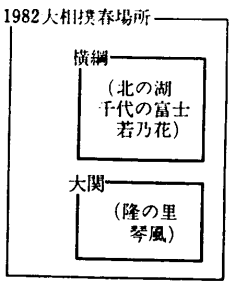

図 8 1982大相撲春場所の概念のフレーム表現
Fig. 8 Representation of the concept of 1982-Sumo-wrestling-spring-tournament by the frame.

次に，内包による表現について考える．内包は，ある概念に固有の属性を表すので，その一部はフレーム表現におけるスロットの表現と等価であると解釈できる．たとえば**図5**に示すような，人の概念の H-net による表現は，**図6**に示すようなフレーム表現と等価であると解釈できる．

最後に，世界について考える．世界は，知識の適用範囲を定めるので，これをそのままフレームと解釈することができる．たとえば**図7**に示すような 1982 年大相撲春場所の世界の H-net による表現は，**図8**に示すようなフレーム表現と等価であると解釈できる．

structure node による表現が，もともと存在しなかった実体が，既存の実体を組み合わせることにより，実体として出現したことを表しているのに対して，内包による表現は，もともと存在する実体の属性を規定している．さらに，世界による表現は，同じ名前の概念が，世界によって異なる外延と内包をもつことを表している．フレーム表現では，異なるフレームの同じ名前のスロットが，実は同一のものであるなどという規定はすべて手続きを付加することにより表現するのに対して，H-net では，上記の3種類の表現を組み合わせることにより自然な形で表現することができる．

最後に，知識表現として重要な，default, ISA hierarchy, inheritance の概念を，H-net 表現に見いだすための解釈を示す．

内包と外延の組合せによる表現を，default 値の表現として解釈することができる．たとえば，日付という概念を**図9**のように表現すれば，これは year の default 値が 1982 であることを表現していると解釈できる．日付は year, month, day からなる structure であることを内包で示し，year の代表値として 1982 を外延で示していると解釈できるからである．この解釈は先に述べたとおりに deduce という関数を定義

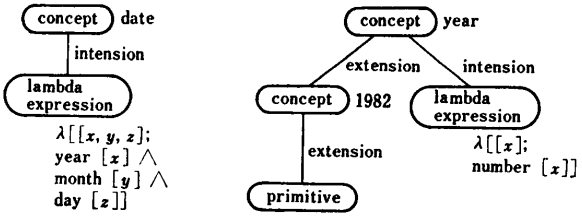

図 9 default 値の表現
Fig. 9 Representation of default value.

することにより成立する．structure 以外の表現についても，値の default は内包と外延の組合せにより表現できる．しかし，「ふつう，鳥は飛ぶ．」というような，述語の真偽の default の表現は H-net だけではできない．換言すれば，リンクで接続されるノードが何であるかの default は表現できるが，リンクのあるなしの default は表現できない，これは純粋なフレーム表現と同様である．

ISA hierarchy は概念と概念が，集合と部分集合あるいはクラスとサブクラスの関係にあることから生まれる階層構造である．知識表現では，ISA hierarchy に沿って，上位の概念の属性が，下位の概念に遺伝するという inheritance の規則を同時に定めることがその記述力を高めるために重要である．この規則が設けられていれば，すべての概念について，属性を細かく記述しなくとも，共通の属性は上位の概念にまとめて記述すればすむし，既存の概念に新しい属性を付け加えるだけで新しい概念を容易に定義できる．この ISA hierarchy と属性の inheritance を H-net においては内包として自然に表現していると解釈できる．たとえば，図5に人の概念の表現を示したが，男の人という概念は**図 10** に示すように表現することができる．図 10 の表現は，男の人というのは，人でありかつ性別が男である実体であるということを内包を用いて表

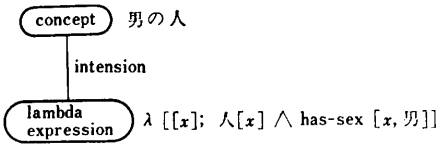

図 10 男の人の概念の表現

Fig. 10 Representation of the concept of male.

現している．これは，男の人が人の部分集合であることを示していると解釈できる．さらに，男の人の内包を評価すれば，自動的に，名前と年齢と誕生日をもつという，人の内包も評価されるので，inheritance も実現される．従来の知識表現では，遺伝すべき性質と遺伝すべきでない性質の区別が問題となるが，H-net においては，内包として表現される性質と，外延として表現される性質とで，これを自然に区別できる．

4. H-net の解釈・操作言語 HNRL

HNRL (H-net Representation Language) は，知識表現 H-net の解釈を行ったり，表現の編集，蓄積，検索などの操作を行ったりするための言語である．現在，その第1版が東京大学大型計算機センタの Vax 11-Unix 上に，Franz Lisp を用いて実装されている．H-net 表現に対するすべての操作は，Lisp の関数として用意されているので，ユーザは Lisp の環境下で H-net の操作を行う．

HNRL の根幹をなす二つの関数は，前章で述べた演繹的推論を行うための関数 deduce と帰納的推論を行うための関数 induce である．

関数 deduce のアルゴリズムは，プログラム言語 Prolog における論駁のアルゴリズムとまったく同様である．すなわち，top-down, depth-first の推論を行い，失敗したらバックトラックする．

関数 induce は try-apply と match という二つの関数からできているが，現在実装されている HNRL では，それぞれのアルゴリズムとしては，原始的なアルゴリズムを採用している．すなわち，try-apply はユーザの指定する既知の知識を，新しいノードに対して，あらゆる順列組合せで適用して，そのノードの内包を推定する．match は，いくつかのノードに対して try-apply が出力した内包どうしの一致する部分を全数検査して調べる．ただし，変数に対しては名前を変えることにより一致可能かどうかの配慮がなされる．問題に応じて，内包の，ある部分に対する重みづけなど，さまざまなヒューリスティックな手法を induce という関数に導入することをユーザが要求するものと思われるので，最も単純なアルゴリズムの関数のみを用意した．H-net においては，事実と原理とその間の関係が明確なので，基本的な関数だけ用意しておけば，さまざまなアルゴリズムの induce が容易に作れるであろう．実際，現在までに，ユーザが順列組合せの一部の排除を指定することのできる try-apply，弁証法的に，矛盾を止揚することにより一致を考える match-aufheben なども作成されている．

現在の HNRL には，H-net をフレームとして解釈し，FRL などの言語と H-net 表現とのやりとりをする関数は実装していないが，前章に述べた解釈により，実装可能である．

その他の，HNRL のおもな関数は，add-concept, add-extension, add-intension, assert-primitives, change-world, parent-child, path-walk などである．add-concept は concept node とその外延と内包の追加を行うための関数である．add-extension は外延の追加を行うための関数である．add-intension は内包の追加を行うための関数である．assert-primitives は primitive node 作成のための関数である．change-world は知識を取り扱う世界を変えるための関数である．parent-child は世界の上下関係を宣言するための関数である．path-walk はリンクをたどるための関数である．HNRL ではリンクは双方向にたどれるように実装してあるので，連想の記述などに用いることができる．また，structure の外部表現として，[ ]，を，set の外部表現として，{ }，の使用を許している．概念の内包を表すための，ラムダ表記された式のなかには，H-net 上に定義された概念だけでなく，任意の Lisp の関数も書けるように実装されている．この機能は主として入出力の記述などに用いられる．ノードがどの世界中に定義されたかの区別は，ノードの内部名に世界の名前を特殊記号を介して連結することにより行っている．世界の階層構造の情報は HNRL 上に存在する．

5. H-net の応用

H-net は当初，自然言語理解システムの作成を念頭において考案された．筆者らは先に，論文抄録の文章の深層構造にいくつかの定型があることに着目し，これをフレームで表現し，フレームに基づき文章を理解するシステムを作成し，実験を行った[8]．このフレームの作成は人手によるため多大な労力を必要とした．

これを，HNRL を用いて機械的に援助することを考える．

これは，Schank らの，スクリプトを文章に適用することにより文章を理解する SAM (script applier mechanism)[9] とは逆に，文章からスクリプトを抽出する SEM (script extractor mechanism) を作成するものである．SEM はまず同じテーマについて書かれたいくつかの文章を構文解析する．次に，いくつかの構文解析結果に対して，HNRL の関数 try-apply を用いて，それぞれの文章において重要と思われる記述を抽出する．その際，SEM においては，構文解析結果を少しずつ参照し，重要な記述か否かをユーザと対話しながら定めるための知識を H-net により作成しておき，この知識と構文解析結果を try-apply の入力とする．try-apply の出力として得られた情報どうしのマッチングを，HNRL の関数 match を用いて調べることにより，あるテーマの文章に共通して記述される内容の定型が抽出される．すなわち，SEM においては，帰納的推論をすべて自動的に実行するわけではなく，人間との対話を行いながら推論を進めてゆく．むろん，このようなシステムは Lisp で直接作成することも可能ではあるが，H-net を用いることにより，情報の整理を容易化することができる．現在，文献コンサルテーションシステムと人間との自然言語による対話を対象に，上記システムの実験を行っているが，その詳細は別稿としたい．

上述したような自然言語理解システム以外にも，代数の定理の発見，ロボットの動作の公理の決定，化学物質の合成法の発見，文字列の性質の発見などの応用に H-net は有用であると考えられる．

数学的応用の例として，2 進数の足し算について，1 桁の足し算の外延をあらかじめ与えておき，2 桁の足し算の外延をいくつか与えることにより，2 桁の足し算の内包を推論するという，簡単な実験を行った．ただし，try-apply に対しては，2 桁の足し算の外延に対して，やみくもに 1 桁の足し算を適用するのではなく，各桁ごとの足し算を行い，その後，どのような関係が成立しているかについて，あらゆる可能性を調べるようにユーザから指示を与えた．この結果，2 桁の足し算の四つの例を与えることにより，CPU time 約 40 分で，くり上がりの手続きを発見し，2 桁の足し算の正しい内包が得られた．

6. むすび

従来のプログラム言語の Lisp や Prolog と，知識表現のフレームやセマンティック・ネットワークの中間に位置する新しい知識表現 H-net を提案した．H-net は 1 階述語論理に理論的基礎をおいているが，知識を表現するのに必要な能力をもっている．H-net の名の由来は，1 階述語論理のモデルに，外延的拡張と内包的拡張により階層構造を導入して，ネットワーク表現としたことにある．

H-net は従来のフレームやセマンティック・ネットワークに比べて，きわだって表現能力が増しているわけではない．むしろ，従来のフレームやセマンティック・ネットワークなどの知識表現を見直し，論理的に明解な構成要素を積み上げて知識表現を行うための，道具として H-net は有用であると考える．セマンティック・ネットワークは種類が多いため，本論文では主として H-net とフレームの関係のみをとりあげたが，H-net とセマンティック・ネットワークの関係も同様に論ずることができる．

H-net は外延と内包という形で事実と原理を区別するので，帰納的推論の記述に有用である．H-net はあらゆる問題について帰納的推論のアルゴリズムを与えるものではない．正しい帰納的推論の可能性やアルゴリズムの研究は本論文とは別の問題である．しかし，従来，帰納的推論の問題に属する問題が，それぞれ問題に応じて別々のプログラムを作成することにより解決されてきたのに対して，知識を整理する共通の土俵を与えるという意味で，H-net は有意義であると考える．

今後，使いやすさに重点をおいて HNRL の改善を行い，自然言語理解システム作成などの実験により，評価を行ってゆきたいと考えている．

謝辞 日頃ご討論いただく，人工知能の輪講会 AIUEO の会員諸氏ならびに猪瀬・斉藤研究室の諸氏に感謝いたします．

参考文献

1) 田中幸吉：知識ベースとその応用，情報処理，Vol. 21, No. 12, pp. 1231-1241 (1980).
2) Shapiro, E. Y.: Inductive Inference of Theories from Facts, *Research Report*, No. 192, Yale University (1981).
3) Winston. P. H.: Learning and Reasoning by Analogy, *Comm. ACM*, Vol. 13, No. 12, pp.

689-703 (1980).
4) Charniak, E.: A Common Representation for Problem-Solving and Language-Comprehension Information, *Artif. Intell.* No. 16, pp. 225-255 (1981).
5) Chang, C. L. and Lee, R. C.: *Symbolic Logic and Mechanical Theorem Proving*, p. 331, Academic Press, New York (1973).
6) Chang C. C. and Keisler, H. J.: *Model Theory*, p. 550 North-Holland, Amsterdam (1973).
7) Woods, W. A.: What's in a Link: Foundation for Semantic Networks, in Bobrow, D. G. and Collins, A. (eds.): *Representation and Understanding*, p. 427, Academic Press, New York (1975).
8) 堀, 斉藤, 猪瀬: シナリオを用いて構造化されたキーワードをアブストラクトから抽出する一手法, 情報処理学会計算言語学研究会資料, 25-2 (1981).
9) Schank, R. C. and Abelson, R. P.: *Scripts, Plans, Goals and Understanding*, p. 248, John Wiley and Sons, New York (1977).
10) 堀, 斉藤, 猪瀬: 帰納的推論も記述容易な知識表現の提案, 情報処理学会昭和57年前期全国大会論文集, pp. 849-850 (1982).

(昭和57年4月 9 日受付)
(昭和57年7月12日採録)