

## 知識を用いた説明テキストの理解と情報抽出†

西田 富士夫<sup>††</sup> 高松 忍<sup>††</sup>  
谷 忠 明<sup>††</sup> 日下 浩 次<sup>††</sup>

説明的テキストは、デバイスやシステムなどのオブジェクトの特性や機能を、専門分野の基本的な知識に基づいて記述している。その説明の内容は、対応する専門分野の一般的な知識を参照して、説明のプロセスをオブジェクトレベルで検証することにより完全に理解することができるものと思われる。本論文は、知識ベースの推論規則を用いて説明のプロセスを検証し、これによって説明的テキストを理解する方法を与えている。対象とするテキストとして論理回路に関する特許文献の詳細説明のテキストを例にとっている。本理解システムは、テキストの内部表現を構文的、字句的に標準化し、論理的推論により理由づけの説明を検証し、テキストにおけるコヒーレントな文間の関係を通してデバイス全体の機能を明らかにしている。解析した結果は、与えられた質問に対しテキストの関連する部分へ速やかにアクセスするのに便利のように関係データベースに蓄積する。

### 1. ま え が き

近年、言語処理や知識工学の発展に伴い、言語や知識処理の研究が活発に行われている<sup>2),5),6),9),10)</sup>。言語理解に関しては物語や談話理解などの研究が進み<sup>1),4)</sup>、知識処理に関しては、プロダクションシステムのような質問回答系の研究が盛んである<sup>3)</sup>。

一方、情報、知識の収集や伝達は従来からテキストやドキュメントの形で行われている。これらのテキストはなんらかの基本的または専門的な一般的な知識に基づいた記述や説明を含んでいる。科学技術などの分野ではこれらの知識は法則や定理であったり、要素やデバイスの機能であったり、これらの製作に関する know how である。これらの分野のテキストは、これらの知識に基づいて、対象とするシステムやデバイスの構成を記述し、前提条件から動作原理や解析や機能やプロセスなどを詳しく説明したり理由づけていることが多い。読み手は、テキストに記述されている事象の系列に対応する知識ベースの部分特定し、これらを具象化し、知識ベースに基づいた推論によって具象化した事象間の原因結果の連鎖としてテキストの事象の系列を捉える。これによりテキストの文のコヒーレンスを理解し、必要とする情報や知識を抽出するものと考えられる。このような見地から、この論文は、システムやデバイスの機能について述べた説明テキ

ストを対象とし、知識ベースを用いた推論によってテキストを理解するとともに、質問に回答するシステムの構築について考察した。

本論文では、テキストやドキュメントからの情報抽出の半自動化を目的として、特許公報に現れる論理回路のテキストを対象として選んだ。これらの分野の知識体系はいわゆる well defined system であるが、新しい提案が多様な自然言語的表現によりどのように説明され、知識ベースによりどのように理解できるかを調べることは重要であると思われる。

本論文の主題は次の3点である。

(1) 知識ベースを援用し推論によってテキスト理解を行うには、テキストに現れる多様な自然言語の内部表現を知識ベースと同じ記述形式の形式表現に変換し、推論を能率的に行うことが望ましい。しかし、従来の研究は主に知識表現や推論に重点がおかれ、多様な自然言語の内部表現を推論操作が可能な形式表現に変換することについてはあまり研究が進んでいない。本論文では、デバイスの動作の因果関係やデバイスの構成などの記述に対して一つの標準的な形式表現を設定し、この形にテキストの内部表現を変換する方法を述べている。以下、この変換を標準化とよぶ。

(2) 質問回答系などに用いられている推論法と知識ベースを導入し、(1)でえられたテキストの形式表現を用いて、その記述内容をオブジェクトレベルで詳しく追跡検証し、説明の本を構成することによりテキストを理解する方法を述べている。

(3) (2)でえられたテキストの解析結果をどのようにデータベース化しておけば、同種の他の事物と特

† Knowledge Based Expository Text Understanding and Information Extraction by FUJIO NISHIDA, SHINOBU TAKAMATSU, TADAOKI TANI and HIROJI KUSAKA (Department of Electrical Engineering, College of Engineering, University of Osaka Prefecture).

†† 大阪府立大学工学部電気工学科

徴を比較でき、説明事項に関しても効率的にテキストの関連部分にアクセスでき回答できるかの問題について、関係データベースを用いる手法について考察している。

## 2. 基本デバイスの動作の形式的表現

テキストに含まれるシステムやデバイスなどの説明文を専門分野の一般的知識と照合し理解するために、基本デバイスに関する知識ベースを構築する。デバイスの機能は、デバイスへの入力信号やデータがデバイスによりどのように処理され変換されて出力されるかを示す入出力関係により表される。論理回路の場合には、入出力に関して、述語論理表現や命題論理表現や真理値表などを用いて記述される。本論文では、従来用いられている命題論理表現<sup>3)</sup>と基本的には同じであるが、テキストの自然言語表現との間の変換を容易にしかつ推論操作が可能ないように、次のような文の格構造に基づく述語論理的表現を導入する。ここで、引数部の OBJECT, LOCATION, VALUE などは各引数の役割名(ラベル)を示す。

```

DEVICE : d
INPUT-OUTPUT RELATION :
  GIVEN (OBJ : vo1, LOC : output1) &...
  & GIVEN (OBJ : vom, LOC : outputm)
  & IN-STATE (VAL : vns)
:- GIVEN (OBJ : vi1, LOC : input1),...,
  GIVEN (OBJ : vin, LOC : inputn),
  IN-STATE (VAL : vcs).      (1)

```

(1)式は、デバイス d において“状態が v<sub>cs</sub> のとき、入力 1 に v<sub>i1</sub>, ..., 入力 n に v<sub>in</sub> が与えられると、状態は v<sub>ns</sub> になり、出力 1 には v<sub>o1</sub>, ..., 出力 m には v<sub>om</sub> が与えられる”という表現に対応している。なお、(1)式の D<sub>1</sub> & ... & D<sub>m</sub> :- C<sub>1</sub>, ..., C<sub>n</sub> の形の表現は、クローズ表現 D<sub>i</sub> :- C<sub>1</sub>, ..., C<sub>n</sub> (ただし、i=1, ..., m) に変換される。(1)式における v の真理値については各組ごとにレコード形式

```

(((vi1, ..., vin), vcs), ((vo1, ..., vom), vns}))      (2)

```

で与える。

デバイスの機能の表現にはこのほか、命題論理式表現、マクロ述語的表現などがある。後者は例えば一致回路や加算回路などのデバイスの機能をマクロに簡単に示すために、(3)式のようにデバイスへの入力の性質名 p や機能名 f を用いてマクロにデバイスの入出力関係を表すものである。

```

GIVEN(OBJ : z(PRED : f, OBJ : x1..xn),
      GOAL : *),
LOC : output)
:- GIVEN(OBJ : x1..xn, LOC : input),
  IN-STATE(VAL : s),
  p(OBJ : x1..xn).      (3)

```

ただし、z の右の (...) は z を修飾・限定する形容詞節的な表現を表す。ここで、\* は z を表し、その表現における z の役割名が GOAL であることを示す。

さて、一般にデバイスやシステムの構成や動作は、基本要素間の接続図を用いて説明されることが多い。これらの接続図は、原理的には、文章情報から分離し<sup>1)</sup>、線図形を構文解析的手法などにより解析し、文と同様に接続述語表現に変換することができる<sup>2)</sup>。ここでは簡単のために、接続図を接続述語表現で与えておくものとする。デバイスがいくつかのまとまった基本デバイスから構成されているときには、知識ベースには次のように、これらの基本デバイス名とそれらの端子間の接続関係を記述し、全体としてのデバイスの機能が説明できるようにしておく。ここで、i<sub>jl</sub>, o<sub>jk</sub> (j=1, ..., n, l=1, ..., n<sub>j</sub>, k=1, ..., m<sub>j</sub>) はそれぞれ、基本デバイス d<sub>j</sub> の入力端子名、出力端子名を表す。

```

DEVICE : d
INPUT-OUTPUT RELATION : .....
CONSTRUCTION :
  COMPONENT :
    DEVICE : d1
      (INPUT : (i11, ..., i1n1),
       OUTPUT : (o11, ..., o1m1))
    .....
    DEVICE : dn
    .....
  CONNECTION
    CONNECT((oik(di), ijl(dj)), .....))
    .....      (4)

```

以下に、デバイスの入出力関係のマクロ述語的表現を中心に二、三の例を示す。

### 例 1

```

DEVICE NAME : RS フリップフロップ
INPUTTERMINAL-INPUT :
  {R入力端子-vir, S入力端子-vis}
OUTPUTTERMINAL-OUTPUT : 出力端子-vo
INPUT-OUTPUT RELATION :

```

GIVEN(OBJ:  $v_o$ , LOC: 出力端子)

&IN-STATE(VAL:  $v_{ns}$ )

:- GIVEN(OBJ:  $v_{ir}$ , LOC: R入力端子),

GIVEN(OBJ:  $v_{is}$ , LOC: S入力端子),

IN-STATE(VAL:  $v_{cs}$ ).

入出力関係における  $v$  の真理値については各組ごとに

$((v_{ir}, v_{is}), v_{cs}), (v_o, v_{ns})$

なるレコード形式で与える.

例 2

DEVICE NAME: アンド

INPUTTERMINAL-INPUT:

入力端子 (1..n)-x(1..n)

OUTPUTTERMINAL-OUTPUT: 出力端子-z

INPUT-OUTPUT RELATION:

GIVEN(OBJ: z, LOC: 出力端子) & z=L

:- GIVEN(OBJ: x(1..n), LOC: 入力端子 (1..n)),

$i \in (1..n), x(i)=L.$  ①

GIVEN(OBJ: z, LOC: 出力端子) & z=H

:- GIVEN(OBJ: x(1..n), LOC: 入力端子 (1..n)),

$i \in (1..n), \text{all}(i) x(i)=H.$  ②

ここに、①、②は入出力関係のマクロな述語表現であるが、①は入力信号が一つでもLレベルであるなら出力がLレベルとなることを示す。

例 3

DEVICE NAME: エンコーダ

INPUTTERMINAL-INPUT:

入力端子 (1..2<sup>n</sup>)-x(1..2<sup>n</sup>)

OUTPUTTERMINAL-OUTPUT:

出力端子 (1..n)-z(1..n)

INPUT-OUTPUT RELATION:

GIVEN(OBJ: z(1..n), LOC: 出力端子 (1..n))

& BINARY-NUMBER (z(1..n))=m

:- GIVEN(OBJ: x(1..2<sup>n</sup>), LOC:

入力端子 (1..2<sup>n</sup>),

$m \in (1..2^n), x(m)=H,$

$i \in (1..2^n), i \neq m, \text{all}(i) x(i)=L.$

上式は、エンコーダの  $m$  番目の入力信号だけが H レベルになると  $m$  の 2 進数が出力されることを表す。

### 3. 標準化

与えられたテキストを理解し処理するために、テキストの各文を構文解析し格構造形式の内部表現を作る。構文解析は、動詞構文パターンとその格構造フレームを用いる並列上昇型の構文解析法<sup>8)</sup>によって行

う。しかし、構文解析によってえられる内部表現は、なお表層表現に依存する部分が多く多様であり、このままでは知識ベースの推論規則が適用できないことが多い。したがってテキストの内部表現を、知識ベースと同じ記述形式の形式表現に変換し、これに基本デバイスの動作などに関する知識を適用することが必要である。

デバイスの動作や構成などの記述に対し、式(1)、(3)、(4)に与えたような表現を標準的な形式表現とする。すなわち、基本述語式:

GIVEN(OBJ: x, LOC: t) ...入出力動作

IN-STATE(OBJ: d, VAL: v)

...デバイスの状態

p(OBJ: x) ...信号の性質

や

CONNECT(SO: t1, GO: t2) ...接続関係

から論理記号  $\neg$ , &,  $\vee$ ,  $\rightarrow$  を用いて構成される表現を標準的な形式表現とする。

ここでは、論理デバイスの動作に関する表現を中心として、形式表現からいろいろな文の内部表現が作られ、これらの内部表現から表層表現が生成されるものと考え、次のような変換規則を与える。そして、規則右辺のいろいろな文の内部表現を規則左辺の一つの形式表現に変換して標準化する。ここで、 $e_i$  ( $i=1, 2$ ) は上述の基本述語式を示し、 $K_i: t$  ( $i=1, 2$ ) は格ラベルとタームの対を表す。

$e_1 \rightarrow e_2$

::= (PREMISE:  $e_1$ , CONSEQUENCE:  $e_2$ ) (5.1)

$e_1$  &  $e_2$

::= (CAUSE:  $e_1$ , RESULT:  $e_2$ ) (5.2)

|(MEANS:  $e_1$ , PURPOSE:  $e_2$ ) (5.3)

|(CONCESSION:  $e_1$ , RESULT:  $e_2$ ) (5.4)

(PRED:  $p_1$ ,  $K_1: t$ ) & (PRED:  $p_2$ ,  $K_2: t$ )

::= (PRED:  $p_2$ ,  $K_2: t$ (PRED:  $p_1$ ,  $K_1: *$ )) (5.5)

|(PRED:  $p_1$ ,  $K_1: t$ (PRED:  $p_2$ ,  $K_2: *$ ))

GIVEN(OBJ: s, LOC:  $t_1$ )

& (PRED: p, AGENT:  $t_2$ )

::= (PRED: させる,

AGENT-信号: s (OBJ: \*, LOC-端子:  $t_1$ ),

OBJ: (PRED: p, AGENT:  $t_2$ )) (5.6)

(5.1), (5.2), (5.3), (5.4) は前提格, 原因格, 手段

格 | 目的格, 譲歩格の副詞節を含む複文による表現であり, それぞれ,

“e<sub>1</sub> すると e<sub>2</sub> する”

“e<sub>1</sub> するので e<sub>2</sub> する”

“e<sub>2</sub> するように e<sub>1</sub> する” または “e<sub>1</sub> することにより e<sub>2</sub> する”

“e<sub>1</sub> {しても | するが} e<sub>2</sub> する”

を表している.

(5.5)は連体修飾節を含む表現で

“p<sub>1</sub> する t を (が) p<sub>2</sub> する”

である.

(5.6)は(5.2)の表現を簡略化して補文を含む表現で表したもので

“端子 t<sub>1</sub> の信号 s が t<sub>2</sub> に p させる”

の形をとる.

式(5)の規則右辺に与えた内部表現は, “与える” や “なる” などの標準的な動詞の格構造の表現である. これと等価で形の異なる動詞や動作名詞を用いた字句表現に対しては, 単語辞書に, 対応する標準的な動詞と格構造への変換規則を記載しておき, これを用いて標準的な動詞に変換する. デバイスなどの説明文に現れる動詞は次のように分けられる.

(1) 入出力動作, (2) デバイス間の転送動作, (3) デバイスと端子の状態 (変化), (4) デバイスの動作  
以下に変換規則の例を示す.

(1) 入出力動作

“(～を～に) 導く / 出力する / 入力する”

(PRED: 導く / 出力する / 入力する,

OBJ-信号: s, GO-端子: t)

=> GIVEN(OBJ: s, LOC: t) (6.1)

(2) デバイス間の転送動作

“(～を～から～に) 送る”

(PRED: 送る,

OBJ-信号: s, SO-端子: t1, GO-端子: t2)

=> GIVEN(OBJ: s, LOC: t1)

& GIVEN(OBJ: s, LOC: t2) (6.2)

(3) 端子の状態 (変化)

“(～が～に) なる”

(PRED: なる, OBJ: 出力 (OBJ-デバイス: d, ATTR: \*), GO-信号: s)

=> GIVEN(OBJ: s, LOC: 出力端子 (d))

(6.3)

(4) デバイスの動作

“(～を) セットする”

(PRED: セットする,

OBJ-フリップフロップ: d)

=> GIVEN(OBJ: H, LOC: 出力端子 (d))

(6.4)

“(～を) オンする”

(PRED: オンする, OBJ-ゲート: d)

=> GIVEN(OBJ: H, LOC: 出力端子 (d))

(6.5)

以上の複文や単文に関する標準化のほかに, 副助詞を含む内部表現の標準化がある. 例えば, 副助詞 “だけ” を含む文 “s 中の t だけが (を) p する” の内部表現:

(PRED: p, K: t(OBJ: \*, COMPOSITE: s)

(OBJ: \*, MOD: 限定))

(7.1)

に推論則を適用するため, 上式を

t ∈ s, (PRED: p, K: t),

r ∈ s, r ≠ t, all(r) ⊃ (PRED: p, K: r) (7.2)

の内容の連言に変換する. ここに, (7.1)の\*はtを表し, 格Kのタームはtがsに属し, かつただ一つに限定されることを表す連体修飾句の述語表現である. また, 副助詞 “のみ”, “ばかり” を含む表現についても “だけ” の場合と同様に式(7.2)の形に変換する. なお, “も” や “たり” などの他の副助詞については多くの場合読み手の注意を促したり理解を助けるもので, ここでは連言関係の論理的表現に変換し他の意味は無視するものとする. また, 格ラベルに関連する “原因” や “手段” などの名詞語や “有する”, “引き起こす” や “用いる” などの動詞語を含む連体修飾句については文献(8)と同様な手段により標準化を行って形式表現に変換する.

以上の処理のほかに, 代名詞などの照応語, 省略, 接続詞 (相当語) の処理がある. 照応語と省略語の処理については, 従来のように前文や前々文の語句のタームから, 格フレームの意味的カテゴリ条件などをみたすものをいくつか先行詞の候補とする. そして, 4章で述べるように, 検証のステップで推論規則と論理的に整合するものを候補から同定し推論を行う. 検証後においては, 同定した代名詞や省略語を形式表現などに復元しておく. また, “その結果, ...” や “このため, ...” などの接続詞 (相当語) については, 構文解析で結果文のマークだけを付け, 原因となる文の同定は検証の段階で行う. 本論文で対象とする説明テキストは, 文間のコヒーレントな関係が主に事象間

の因果関係であるような well defined system に関するものである。このような well defined system では、因果関係を明確な形の推論規則で記述することができ、これらの規則を用いた検証により、5章の検証例に示すように、このような範囲の説明テキストに現れる照応語や省略語を多くの場合同定し復元することができる。

#### 4. 推論のチェックと記述の整備

与えられたテキストの内部表現の標準表現に基づいて、理解システムはテキストに書かれている各デバイスの動作を順次チェックする。すなわち、テキストに記載のように、デバイスの構成表や接続表や入力条件から出力条件が成立するかどうかを知識ベースを用いた導出形を求める推論により検証する。推論は論理デバイスの場合、テキストに省略のある場合でも通常数ステップの程度である。検証の後、説明文の前提結果の連鎖から説明の木を作り、テキスト内容をまとめる。従来、このような検証手順は明確な形で与えられていない。本論文では、以下の検証手順と次章の検証例に示すように、説明テキストの構造に沿って推論を行い、知識ベースの規則またはその具象例によりテキストの省略部分を補完し検証する方法を与えている。

テキストの説明文は目的やあらましなど導入部やデバイスの構成の記述に続き、前提文、前提結果文、結果文などの繰り返しからなる。前提文  $F_0$  はデバイスの構成や接続条件、信号入力などについて記述する。前提結果文の前提部は前提文か、またはこれに先行するいくつかの前提結果文の結果部  $F_i$  よりなり、

$$F_i \vdash F_{i+1} \quad (8)$$

の形をとる。ただし、'⊢' は '導く' を表すものとする。結果文は前提結果文の前提部を、これが既に先行文に含まれるなどにより、切り離したもので  $F_{i+1}$  の形をとり、前提文と同じ事実文の構造をとる。

推論によるチェックとは、デバイスの動作などの各部分や各過程に関するテキストの説明を、各基本デバイスの動作規則やテキストで述べた仮定などに基づいてチェックし、これらをつないで、デバイス全体としての入出力関係などの機能を検証するとともに、標準化によりえられているテキストの形式表現列を整備するものである。テキストの説明は各プロセスに分解して行われ、各プロセスの推論は小規模のものであるので、テキストに現れる文の順に前向きに検証を行うものとする。そ

して、(8)式のような各々の前提結果文または結果文の各  $F_{i+1}$  の検証については以下の手順に示すように後向き推論により行う。

検証手順では、テキストの前提や中間結果を保持する前提スタックと、記憶素子などの状態を保持する状態スタックを用いる。始めに、前提スタックにテキストの前提をセットし、状態スタックに各記憶素子などの初期状態の表現をセットする。

(検証手順)

ステップ 1:  $F_{i+1}$  に関係するいくつかのデバイス名、動作名をキーとして、式(1)の入出力論理表現に基づき、真理値表のレコードなどを参照して、 $F_{i+1}$  と単一化できる出力または次の状態が知識ベースに記述されているかどうかを調べる。なければステップ 3 へ、あればステップ 2 へ。

ステップ 2: 関係するデバイスの動作規則やデバイス間の接続関係に基づき、順次、推論規則を適用して、 $F_{i+1}$  を導出する入力条件  $F_i$  が前提文や先行する結果文の集合に含まれているかどうかを前提スタックから調べる。含まれていれば検証されたものとし、推論結果からえられるテキストの省略部分と  $F_{i+1}$  を前提スタックに加える。含まれていなければステップ 3 へ。

ここで、 $F_{i+1}$  を導くプロセスにおいて、記憶素子などに状態変化が生じていれば状態スタック内のその記憶素子の状態表現を更新する。

ステップ 3: 停止してユーザの援助を要求する。

以上の手順によりえられる一段の推論プロセスを図 1 のような図式で記憶する。図 1 は、デバイス  $d$  への入力先行段の出力(1)からデバイス  $d$  の接続関係(3)によりえられ、さらに、 $d$  の入力と現在の状態(2)からデバイスの動作を表す推論規則(4)を用いて、 $d$  の出力(5)と次の状態(6)を前述のような検証プロセスにより推論することができることを示す。

なお、アナフォラ処理で選定された先行詞の候補については、これらを順位の高いものから上述の手順に

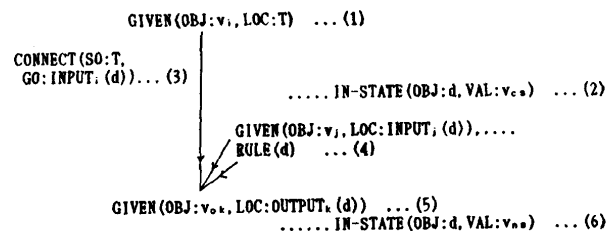


図 1 推論図式

Fig. 1 An inference diagram.

より検証し、論理的に整合する候補を先行詞として同定する。

5. テキスト検証の例

以下にテキスト理解の例を示す。対象としたテキスト<sup>12)</sup>の標題、動作記述の部分と回路図を図2(a)に示す。また、図2(a)の動作記述のテキストを構文解析し、アナフォラ処理と簡単な標準化を行い、えられる内部表現を図2(b)に示す。ここで、②、⑦、⑩の文に含まれる照応語“これにより”に対し、内部表現の{ }内に示すように先行詞の候補が選定される。

次に、動作記述文の検証のため、式(5)や式(6)の規則を用いて知識ベースと同様な表現をもつ図2(c)のような形式表現に内部表現を変換する。ここで、限定の副助詞“だけ”を含む文④の内部表現は式(7.1)→式(7.2)などの規則により図2(c)の④の形式表現に変換され、#3以外の残りのタームの述語についても検証が行われる。

続いて、動作記述文を知識ベースを用いて図2(d)に示すように順次検証する。(d)の検証結果において、各ノードの STATE の欄は、その時点における

状態スタックの内容すなわち記憶素子の状態表現を示す。ただし、記憶素子の状態が先行するノードの状態と同じである場合には省略している。また、(d)の検証結果から⑦、⑩の文の照応語“これにより”はそれぞれ、⑥、⑧を指示することが同定される。なお、②→③、③→④、⑩→⑪の検証では、図1のような推論をそれぞれ2回、3回、5回行っている。

6. 情報の抽出と質問回答

テキストの主要な内容は解析の結果、構成や機能などのサブフレームごとに関係データベースなどに格納すれば、同種のほかのデバイスやシステムと比較や関係検索などを能率的に行うことができる<sup>8)</sup>。動作の説明に関しても同様である。例えば論理デバイスの動作説明を構成要素への入力、出力、適用ルール名などのタプルの系列として関係データベースに格納し、またテキストの対応する詳細記述場所をそれぞれ記録しておく。これにより説明テキストの構造は見やすくなり、質問に対して迅速に関係表や知識ベースやテキストの該当場所にアクセスして推論し回答することができる。

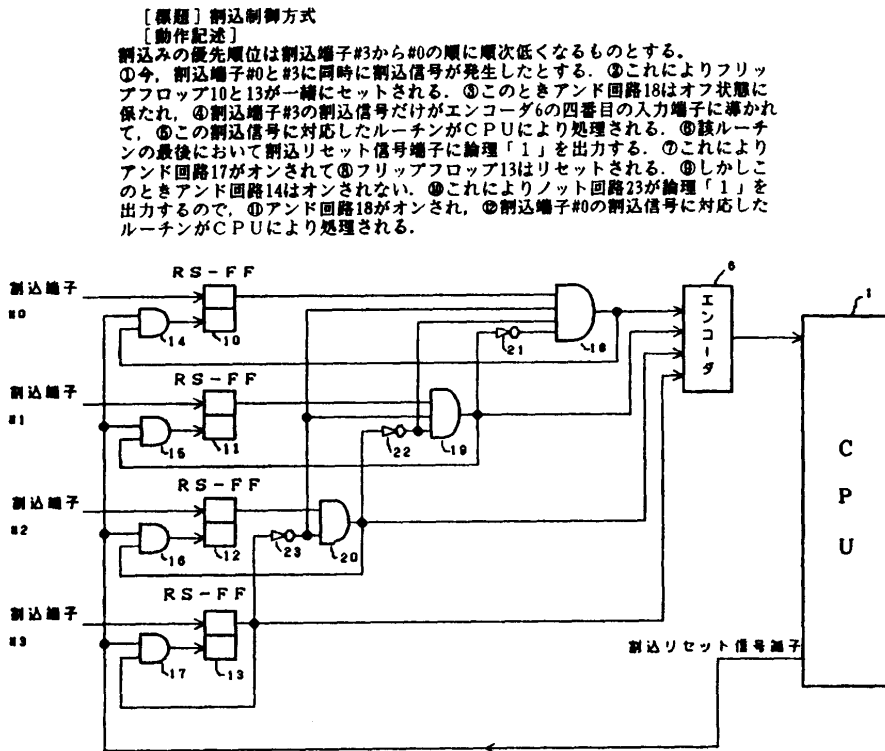


図2(a) テキストと回路図  
Fig. 2(a) A text and the circuit diagram.

さて、図1のように、デバイスの動作の各ステップは、(1)デバイスへの入力やデバイスの入力となる前段入力、(2)現在の状態、(3)デバイス間の接続関係、(4)デバイスの動作に関する(推論)規則、(5)デバイスからの出力、(6)次の状態で表すことができ

- ① (PRED:発生する (MOD:仮定), OBJ:割込信号, LOC:割込端子#0, #3, MANN:同時)
- ② (PRED:セットする (VOICE:受動), OBJ:フリップフロップ10, 13, MANN:一掃, CAUSE: { ① ; 割込信号 in ① })
- ③ [(PRED:保つ (VOICE:受動), OBJ:アンド回路18, LOC:オフ状態, TIME:このとき) AND (PRED:導く (VOICE:受動), OBJ:割込信号 (OBJ:\*, LOC:割込端子#3, MOD:限定), GO:入力端子 (OBJ:\*, POSSESSOR:エンコーダ6, ORD:4)) AND (PRED:処理する (VOICE:受動), AG:C P U, OBJ:ルーチン (PRED:対応する, OBJ:\*, PARTIC: 割込信号 (OBJ:\*, LOC:割込端子#3)))]
- ④ (PRED:出力する (VOICE:受動), OBJ:ルーチン (PRED:対応する, OBJ:\*, PARTIC: 割込信号 (OBJ:\*, LOC:割込端子#3)))
- ⑤ [(PRED:オンする (VOICE:受動), OBJ:フリップフロップ, CAUSE: { ④;論理「1」 in ④;⑤;...}) AND (PRED:リセットする (VOICE:受動), OBJ:フリップフロップ13)]
- ⑥ (PRED:オンする (ESS:否定, VOICE:受動), OBJ:アンド回路14, TIME:このとき) [CAUSE: (PRED:出力する, AG:ノット回路23, OBJ:論理「1」, CAUSE: { ④;⑤;...})] RESULT: (PRED:オンする (VOICE:受動), OBJ:アンド回路18) AND (PRED:処理する (VOICE:受動), AG:C P U, OBJ:ルーチン (PRED:対応する, OBJ:\*, PARTIC: 割込信号 (OBJ:\*, LOC:割込端子#0)))]

図 2(b) 内部表現

Fig. 2(b) The internal expressions.

- ①-1 GIVEN (OBJ:割込信号, LOC:割込端子#0)
- ①-2 GIVEN (OBJ:割込信号, LOC:割込端子#3)
- ②-1 GIVEN (OBJ:H, LOC:出力端子 (RSフリップフロップ10))
- ②-2 GIVEN (OBJ:H, LOC:出力端子 (RSフリップフロップ13)) CAUSE: {①}
- ③ [GIVEN (OBJ:L, LOC:出力端子 (アンド回路18)) & GIVEN (OBJ:X, LOC:入力端子 (エンコーダ6)) X(4)=割込信号 (OBJ:\*, LOC:割込端子#3) i ∈ (1..4), i ≠ 4, all(i) X(i)=L] & 処理する (AG:C P U, OBJ:ルーチン (PRED:対応する, OBJ:\*, PARTIC: 割込信号 (OBJ:\*, LOC:割込端子#3)))]
- ④ GIVEN (OBJ:H, LOC:割込リセット信号端子)
- ⑤ [GIVEN (OBJ:H, LOC:出力端子 (アンド回路17)) CAUSE: {④;⑤;...}] & GIVEN (OBJ:L, LOC:出力端子 (RSフリップフロップ13))]
- ⑥ GIVEN (OBJ:L, LOC:出力端子 (アンド回路14))
- ⑦ [GIVEN (OBJ:H, LOC:出力端子 (ノット回路23)) CAUSE: {④;⑤;...}] & GIVEN (OBJ:H, LOC:出力端子 (アンド回路18))
- ⑧ 処理する (AG:C P U, OBJ:ルーチン (PRED:対応する, OBJ:\*, PARTIC: 割込信号 (OBJ:\*, LOC:割込端子#0)))]

図 2(c) 形式表現

Fig. 2(c) The formal expressions.

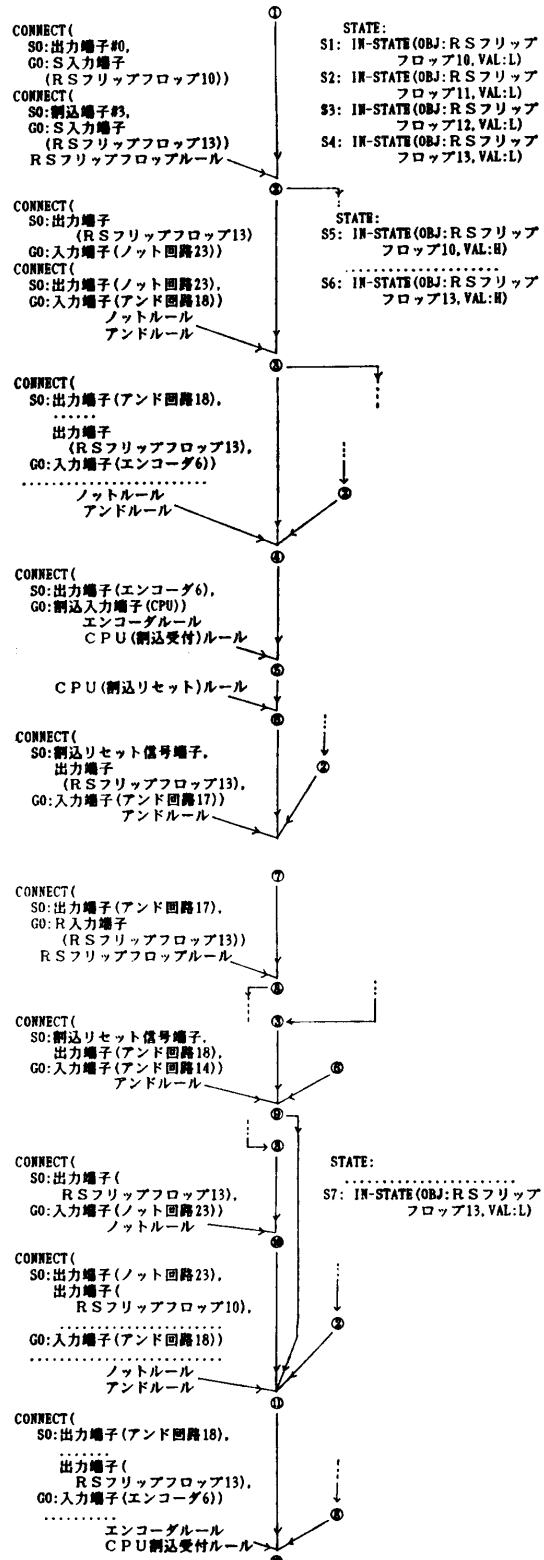


図 2(d) 検証結果

Fig. 2(d) The result of verification.

る。これらの組を一つのタプルとし、これらのタプルの系列として図3の(1)のような関係表の形に抽出する。

図3の(1)は前章の図2(d)の推論過程から抽出した入出力の関係である。また(2)は図2(c)の形式表現から抽出した入出力の内容を示し、(3)は記憶素子の状態の内容を示し、(4)と(5)は回路図からえられる接続情報を示す。

えられた関係表と知識ベースを用いた推論により、テキストの内容のほか他のデバイスとの比較やサブデバイスの置き換えによる小変更などに関する質問に回答することができる。特に、以下の質問回答手順と回答例に示すように、本手法により次のような質問に対し能率的に回答することができる。

(a)テキストの内容に沿った質問、(b)テキストと

(1) PREMISE-CONSEQ-RELATION

PREMISE (1)入力 (2)入力状態	CONSEQ (5)出力 (6)出力状態	CONNECTION (3)接続関係	RULE (4)規則
{①-1, ①-2} S1, S2, S3, S4	{②-1, ②-2} S5, S6	C1, C2	RS7リッブア7ロツア7ル-ル
{②-2}	③	C3, C4	ノットル-ル, フントル-ル
{②-1, ②-2, ③, ...}	④	C5	

(2) INPUT-OUTPUT-RELATION

文番号	PRED	OBJ	LOC
①-1	GIVEN	H	T1
①-2	GIVEN	H	T2
②-1	GIVEN	H	T18
②-2	GIVEN	H	T5
③	GIVEN	L	T9
④	GIVEN	[L, L, L, H]	T10

(3) IN-STATE

ID	OBJ	VAL
S1	RS7リッブア7ロツア710	L
S2	RS7リッブア7ロツア711	L

(4) CONNECTION

ID	PRED	SO	GO
C1	CONNECT	T1	T3
C2	CONNECT	T2	T4
C3	CONNECT	T5	T6
C4	CONNECT	T7	T8
C5	CONNECT	{T9, T5, ...}	T10

(5) TERMINAL-DEVICE-RELATION

ID	TERMINAL	DEVICE
T1	割込端子#0	
T2	割込端子#3	
T3	S入力端子	RS7リッブア7ロツア710
T4	S入力端子	RS7リッブア7ロツア713
T5	出力端子	RS7リッブア7ロツア713

図3 関係データベース

Fig. 3 The relational database.

異なる入力条件の場合の動作に関する質問

質問文は前提結果文の形で与えられ、例えば、あるサブデバイス  $d_i$  の入力条件  $p'$  に対してサブデバイス  $d_j$  の出力(条件)の値を問うものとする。質問文を構文解析し、えられた内部表現を標準化して形式表現に変換する。この形式表現を [SENT-NO:  $n$ , PREMISE:  $p'$ , RESULT:  $q'$ ] とする。ここに、 $n$  は質問に関連するテキストの文の番号であり、 $q'$  または  $p'$  は未知項  $x$  を含んでもよいものとする。以下に手順の概略を示す。

(質問回答手順)

ステップ1: PREMISE-CONSEQUENCE RELATION 表から文番号  $n$  の時点における記憶素子の状態の値を検索し復元する。そしてステップ2へ。

ステップ2: 質問の PREMISE 部  $p'$  に関連する PREMISE-CONSEQ RELATION 表のタプル:

[PREMISE:  $p$ , CONSEQ:  $s$ ,  
CONNECTION:  $c$ , RULE:  $r$ ]

にアクセスし、ステップ3へ。

ステップ3: デバイスルール  $r$  と接続関係  $c$  を用いて、 $p'$  からサブデバイス  $d_i$  から  $d_j$  へ5章と同様な推論を行い、 $s'$  を導出する。

$q'=s'$  なら、停止し推論結果から質問に回答する。

$q' \neq s'$  なら、 $s'$  を  $p'$  とおいてステップ2へ。

サブデバイス  $d_i$  における出力条件  $q'$  を与えて  $q'$  を満たす  $d_i$  における入力条件  $p'$  を求める場合には  $d_i$  から逆向きに推論を行うほうが能率的である。サブデバイスをほかのもので置き換えた場合にも同様に推論を行うことができる。

次に、関係表と知識ベースを用いて質問に回答する例を示す。

質問: 図2(a)の例題の⑤の動作に続いて割込端子 #1 に割込信号が発生したとき、次にどの割込信号に対応するルーチンが CPU により処理されるか。

質問文を解析して、えられた内部表現を次の形式表現に変換する。

[PREMISE: GIVEN (OBJ: 割込信号,  
LOC: 割込端子 #1) ①/  
CONSEQ: 処理する (AG: CPU, OBJ: ルーチン  
(PRED: 対応する, OBJ: \*, PARTIC:  
割込信号 (OBJ: \*, LOC: ?X))) ②/  
]

最初に、⑤の動作に対応する PREMISE-CONSEQ



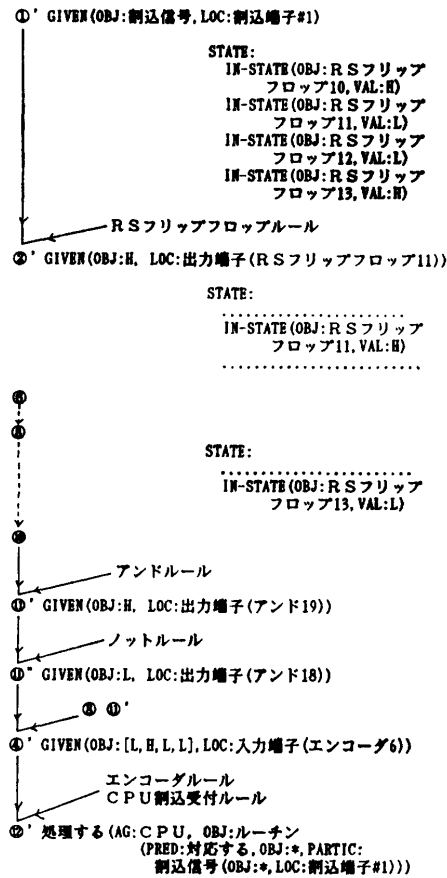


図 4 推論図  
Fig. 4 The inference diagram.

RELATION 表のタプルにアクセスし、その時点の状態表現を検索する。

以下、[PREMISE: ①', CONSEQ: ②'] の推論は、アクセスした表をガイドとして上述の手順により図 4 のようにえられ、割込端子 # 1 の割込信号に対応するルーチンが CPU により処理される。

7. あとがき

図 2 (a) のような論理回路デバイスに関する特許テキストを対象として、構文解析、標準化、検証による理解、関係表への情報抽出と質問回答の計算機実験を行った。システムは LISP 言語で記述し、推論による検証には LISP-PROLOG を用いた。計算機は ACOS-930 を使用し、処理に必要な容量は約 400k バイトである。構文解析から情報抽出までの平均処理時間は、インタプリタモードで約 100 語のテキストに対し約 90 秒であった。標準化の変換規則と知識ベースの推論規則を作成するため、特許テキスト約 30 件の動作記述部を分析し、これらのテキストに対しハンドシミュ

レートし、そのうち約半数程度のテキストを計算機処理した。構文解析における係受け関係のあいまいさについては、格構造フレームのタームの意味カテゴリを専門分野のより詳しい下位のカテゴリを指定することにより解消をはかり<sup>8)</sup>、プリエディットを要する標準的でない文などを除き、テキストの約 8 割程度を解析することができた。えられた内部表現を 3 章で与えた変換規則により形式表現に変換し、等号関係に関する高階論理や必要性・可能性などの記述を除き、一階論理の範囲で約 9 割程度を能率的に検証することができた。検証によるテキスト理解と質問回答については、計算機実験から本方法により能率的に行えることが確かめられ所期の結果がえられた。

ここでは論理回路テキストの理解について述べたが、一般に、推論則を適用するため、テキストの内部表現を目的に変換したり推論する研究が今後とも重要である。電子回路などのアナログ的な機能や動作の定性的な記述についても論理回路の場合と同様な方法により、定電圧回路に関するものなど数件のテキスト理解の計算機実験を行い同様な結果がえられた。例えば、トランジスタなどの素子の特性やオームの法則などの関係を定性的に if-then 文などで知識ベースに記述しておき、テキストの記述と照合すればよい。これらの文書は一般に数式を用いた記述を含んでいるが、これらは数式処理システムなどの導入により理解のための推論処理が同様に可能になるものと思われる、今後の課題である。

謝辞 本研究におけるシステムの作成と実験に協力された石谷高志君 (現在(株)シャープ) に深謝いたします。

参 考 文 献

- 1) Schank, R. C. and Rieger, C. J.: Inference and the Computer Understanding of Natural Language, *Artif. Intell.*, Vol. 5, pp. 373-412 (1974).
- 2) Hobbs, J. R.: Coherence and Interpretation in English Texts, *Proc. of 5th IJCAI*, pp. 110-116 (1977).
- 3) Genesereth, M. R.: The Use of Design Descriptions in Automated Diagnosis, *Artif. Intell.*, Vol. 24, pp. 411-436 (1984).
- 4) Alterman, R.: A Dictionary Based on Concept Coherence, *Artif. Intell.*, Vol. 25, pp. 153-186 (1985).
- 5) Hahn, U. and Reimer, U.: TOPIC Essentials, *COLING-86*, pp. 497-503 (1986).
- 6) 安部, 石川, 辻: 組立説明文からの組立手順の生

- 成, 人工知能学会誌, Vol. 3, No. 5, pp. 590-598 (1988).
- 7) Nishida, F., Takamatsu, S. and Fujita, Y.: Extraction of Structured Information from Texts with Various Forms of Information, *Proc. of Int. Conf. on Text Processing with a Large Character Set*, pp. 256-261 (1983).
- 8) 高松, 日下, 西田: 技術抄録文からの関係情報の自動抽出, *情報処理学会論文誌*, Vol. 25, No. 2, pp. 216-224 (1984).
- 9) Nishida, F., Takamatsu, S., Tani, T. and Kusaka, H.: Text Analysis and Knowledge Extraction, *COLING-86*, pp. 241-243 (1986).
- 10) 高松, 西田: 見出し情報を用いたテキスト解析と情報抽出, *情報処理学会論文誌*, Vol. 29, No. 8, pp. 760-769 (1988).
- 11) 山田, 美濃, 坂井: 画像特徴を利用した文書画像の蓄積・検索方式, 第 33 回情報処理学会全国大会論文集, 6P-7, pp. 1663-1664 (1986).
- 12) 特許公報 昭 54-17617: 割込み制御方式, 日本国特許庁, 分類 97(7)F 13 (1979).

(平成元年3月17日受付)  
(平成2年1月16日採録)

#### 西田富士夫 (正会員)

大正 15 年生. 昭和 25 年京都大学工学部電気工学科卒業. 現在, 大阪府立大学工学部電気工学科教授. 工学博士. 自然言語処理, プログラムの仕様と詳細化, 問題解決システムなどの研究に従事. 著書「言語情報処理」など. 電子情報通信学会, 電気学会, 計測制御学会, IEEE 各会員.

#### 高松 忍 (正会員)

昭和 23 年生. 昭和 46 年大阪府立大学工学部電気工学科卒業. 昭和 48 年同大学院工学研究科修士課程修了. 工学博士. 現在, 大阪府立大学工学部電気工学科講師. 自然言語処理, 知識情報処理の研究に従事. 電子情報通信学会, 人工知能学会, 日本認知科学会各会員.

#### 谷 忠明 (正会員)

昭和 30 年生. 昭和 57 年大阪市立大学文学部卒業. 現在, 大阪府立大学工学部電気工学科教務技師. 自然言語処理の研究に従事.

#### 日下 浩次 (正会員)

昭和 8 年生. 昭和 33 年大阪府立大学工学部電気工学科卒業. 昭和 35 年同大学院工学研究科修士課程修了. 現在, 大阪府立大学工学部電気工学科助教授. 工学博士. FDM-FM, FSK および SS 通信方式, デジタル信号の伝送と処理などに関する研究に従事. 電子情報通信学会, IEEE, テレビジョン学会各会員.